PB86-146529

Microcomputer Programs for Particulate Control Documentation

Research Triangle Inst.
Research Triangle Park, NC

Prepared for

Environmental Protection Agency
Research Triangle Park, NC

Sep 85

U.S. Department of Commerce
National Technical Information Service
NTIS

PB86-146529

EPA/600/8-85/025a
September 1985

MICROCOMPUTER PROGRAMS FOR PARTICULATE CONTROL

By

M. Kathleen Owen
and
Andrew S. Viner

Research Triangle Institute
P.O. Box 12194
Research Triangle Park, North Carolina 27709

Cooperative Agreement No. CR808936

EPA Project Officer: Leslie E. Sparks
Air and Energy Engineering Research Laboratory
Research Triangle Park, NC 27711

AIR AND ENERGY ENGINEERING RESEARCH LABORATORY
OFFICE OF RESEARCH AND DEVELOPMENT
U.S. ENVIRONMENTAL PROTECTION AGENCY
RESEARCH TRIANGLE PARK, NC 27711

REPRODUCED BY
NATIONAL TECHNICAL
INFORMATION SERVICE
U.S. DEPARTMENT OF COMMERCE
SPRINGFIELD, VA. 22161

TECHNICAL REPORT DATA
(Please read Instructions on the reverse before completing)

1. REPORT NO. EPA/SW/DK-85/009a
EPA/600/8-85/025a

2\.

3. RECIPIENT'S ACCESSION NO.
PB86-146529

4. TITLE AND SUBTITLE
Microcomputer Programs for Particulate Control

5. REPORT DATE
September 1985

6. PERFORMING ORGANIZATION CODE

7. AUTHOR(S)
M.K. Owen and A.S. Viner

8. PERFORMING ORGANIZATION REPORT NO.

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Research Triangle Institute
P.O. Box 12194
Research Triangle Park, North Carolina 27709

10. PROGRAM ELEMENT NO.

11. CONTRACT/GRANT NO.
CR808936

12. SPONSORING AGENCY NAME AND ADDRESS
EPA, Office of Research and Development
Air and Energy Engineering Research Laboratory
Research Triangle Park, NC 27711

13. TYPE OF REPORT AND PERIOD COVERED
Final; 1-6/85

14. SPONSORING AGENCY CODE
EPA/600/13

15. SUPPLEMENTARY NOTES AEERL project officer is Leslie E. Sparks, Mail Drop 61, 919/541-2458. EPA-600/8-85-025b identifies a four-disk package complementing the manual (for all except the Dynamic Display Manual). For Diskettes see PB86-146511

16. ABSTRACT

The report describes the development of a set of programs especially for air-pollution related problems. (The phenomenal growth of the microcomputer industry has made it possible for anyone in industry to have a relatively fast and easy-to-use computer. Unfortunately, the specialized software needs of the air pollution community have not been met.) The report describes: microcomputer programs of electrostatic precipitator performance, an implementation of the GCA/EPA baghouse model; an in-stack opacity prediction program; and a plume opacity model. User benefits, hardware/software requirements, and general restrictions are given, as well as brief descriptions of the theoretical models. A sample problem, which gives detailed instructions on how to use the menus and obtain desired results, is included for each program.

17. KEY WORDS AND DOCUMENT ANALYSIS

a. DESCRIPTORS		b. IDENTIFIERS/OPEN ENDED TERMS	c. COSATI Field/Group	
Pollution	Fabrics	Pollution Control	13B	11E
Computer Programs	Filtration	Stationary Sources	09B	
Particles	Opacity	Microcomputers	14G	
Dust	Plumes	Particulate	11G	21B
Aerosols	Flue Gases	Baghouses	07D	
Electrostatic Precipitators			13I	

18. DISTRIBUTION STATEMENT Release to Public	19. SECURITY CLASS *(This Report)* Unclassified	21. NO. OF PAGES 318
	20. SECURITY CLASS *(This page)* Unclassified	22. PRICE $28.95

EPA Form 2220-1 (9-73)

ATTENTION

PORTIONS OF THIS REPORT ARE NOT LEGIBLE.
HOWEVER, IT IS THE BEST REPRODUCTION
AVAILABLE FROM THE COPY SENT TO NTIS.

NOTICE

This document has been reviewed in accordance with U.S. Environmental Protection Agency policy and approved for publication. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

ABSTRACT

The phenomenal growth of the microcomputer industry has made it possible for anyone in industry to have a relatively fast and easy-to-use computer at their disposal. Unfortunately, the specialized software needs of the air pollution community have not been met. In an attempt to address this need the Research Triangle Institute, under contract to the U.S. Environmental Protection Agency (EPA), has developed a set of programs especially for air pollution-related problems. In this report microcomputer programs of electrostatic precipitator performance, an implementation of the GCA/EPA baghouse model; an in-stack opacity prediction program; and a plume opacity model are described. User benefits, hardware/software requirements, and general restrictions are given as well as brief descriptions of the theoretical models. A sample problem, which gives detailed instructions on how to use the menus and obtain desired results, is included for each program.

CONTENTS

Chapter Page

ABSTRACT iii
FIGURES vii
TABLES x

1 COMPUTER PROGRAMS FOR PARTICULATE CONTROL TECHNOLOGY . . . 1
Introduction 1
Program Menus 2

2 ESP SECTION FAILURE MODEL 6
Revision 1.0 6
Abstract 6
User Benefits 6
Hardware/Software Requirements 6
Method 7
General Restrictions 10
Sample Problem 12

3 ESP DYNAMIC DISPLAY MODEL 62
Revision 1.2 62
Abstract 62
User Benefits 62
Hardware/Software Requirements 62
Method 63
General Restrictions 69
Sample Problem 69

4 GCA/EPA BAGHOUSE MODEL 139
Revision 1.2 139
Abstract 139
User Benefits 139
Hardware Software Requirements 140
Method 139
General Restrictions 141
Sample Problem 141
References 154

5 PLUME OPACITY PREDICTION MODEL 185
Revision 1.0 185
Abstract 185
User Benefits 185
Hardware/Software Requirements 185
Method 186
General Restrictions 186
Sample Problem 187
References 199

Preceding page blank

CONTENTS (continued)

Chapter Page

6 IN-STACK OPACITY CALCULATOR 226
Revision 1.2 226
Abstract 226
User Benefits 226
Hardware/Software Required 226
Method 226
General Restrictions 229
Sample Problem 229
References 235

Appendix A 262
Appendix B 266
Appendix C 281

FIGURES

Number | | Page
1-1 The Main Menu for the PLUMIN data entry program 3
1-2 A data entry menu from the PLUMIN program 3
1-3 A data entry menu from the dynamic ESP model 5
1-4 Request for data in dynamic ESP model 5
2-1a A hypothetical ESP 11
2-1b ESP with one failed section 11
2-1c Two failed sections in channel 1 11
2-1d Two failed sections; one in each channel 11
2-2 The Main Menu for the ESP section failure program 13
2-3 The printer choices menu 13
2-4 Data entry menu 15
2-5 Results for the normal case 17
2-6 "Input Section Failures" menu 17
2-7 Results for the "Calculate Failure Effects" option 18
2-8 Velocity maldistribution input menu 21
2-9 Results of change in NSD in velocity maldistribution menu . 21
2-10 Computer screen results for "Calculate Velocity Effects" . 22
2-11 Printout of velocity maldistribution effects 23
2-12 Screen response to "Save Data to File" Option 23
3-1 A hypothetical ESP 67
3-2 A typical ESP rapping schedule 68
3-3 Main Menu of ESP dynamic model 71

FIGURES (continued)

Number		Page
3-4	ESP parameters input menu	71
3-5	RUN-TIME parameters input menu	74
3-6	Menu organization	76
3-7	Rapping schemes option menu	77
3-8	Strip-Chart display	79
3-9	"Instantaneous" values of mass emissions	80
3-10	Averaged mass emissions	81
3-11	Printer options menu	83
3-12	Strip-Chart printout	85
3-13	"Graph Thickness" printout	86
3-14	"Print NUMERIC VALUES" printout	87
3-15	Static model printout	88
3-16	"Save Data" file name request	88
4-1	The Main Menu of the DATIN program	143
4-2	"HELP" pages called from Main Menu of DATIN program	143
4-3	"Input Design Data" menu	144
4-4	Help routine called from "Design Data" menu	144
4-5	"Design Data" menu after data entry	146
4-6	"Input Operating Data" menu	146
4-7	Page 1 of "Input Dust & Fabric Properties" menu	147
4-8	Page 2 of "Input Dust & Fabric Properties" menu	149
4-9	"Input Special Instructions" menu	149
4-10	Request for file name	151
4-11	Printout from DATIN for sample problem	152

FIGURES (continued)

Number		Page
4-12	Results for sample problem	153
5-1	The Main Menu for the PLUMIN data entry program	188
5-2	Help pages for the Main Menu	188
5-3	"Input Ambient Parameters" menu	189
5-4	Help page for the "Ambient Parameters" menu	189
5-5	Page 1 of the "Input Plume Parameters" menu	191
5-6	Help page called in "Plume Parameters" menu	191
5-7	Page 2 of the "Input Plume Parameters" menu	191
5-8	Response to "Save . . ." or "Read" options of PLUMIN . . .	193
5-9	Printout for PLUMIN program	194
5-10	User input to PLUME program	195
5-11	Output from PLUME program	196
6-1	Title page	230
6-2	Main Menu for instack opacity program	230
6-3	Input data menu	230
6-4a	Entry page for the real part of the refractive index . . .	233
6-4b	Entry page for the imaginary part of the refractive index .	233
6-5	Choose Qext Calculation menu	234
6-6	Input data output for "Integrate . . ." option	234
6-7	Results output for "Integrate . . ." option	234
6-8	Input data output for "Calculate Qext" option	236
6-9	Results output for "Calculate Qext" option	236

TABLES

Number		Page
3-1	Explanation of Variables Used in Data Menus	72
6-1	Units for Input Data	231

CHAPTER 1

COMPUTER PROGRAMS FOR PARTICULATE CONTROL TECHNOLOGY

INTRODUCTION

Over the past few years, the Research Triangle Institute (RTI), under contract to the U.S. Environmental Protection Agency (EPA), has developed and/or adapted a number of computer programs for microcomputers to aid in research on particulate control technology. These programs have proven useful in understanding and predicting the behavior of baghouses, ESPs, and stack opacity. The purpose of this report is to provide the necessary information for using the programs as well as some information on the theoretical background of the programs.

The remainder of this report is divided into chapters, each describing a different program. Each chapter follows the same general format. The first section of the chapter includes the program name, revision number, abstract, and a description of the hardware and software required to run the program. In the next section the equations used in the program are described and/or referenced to provide an overview of the calculation method used in the program. After the description of the method, a sample problem is given with a listing of the input parameters and consequent output parameters included. The sample problems include user instructions describing how to run the program as an aid in the execution of the program. The last part of each chapter is a complete listing of the program(s) described in that chapter.

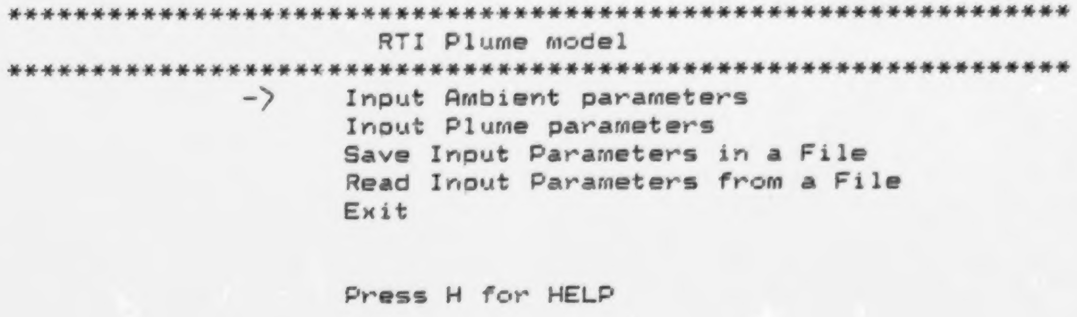
Although these programs were originally written for the TRS-80 (Radio Shack, a division of Tandy Corporation, One Tandy Center, Fort Worth, TX) microcomputer, some have been adapted for use with the IBM PC (IBM Corporation, Boca Raton, FL). The main body of this report is specifically written for use with the TRS-80 versions. However, it may be used with the IBM PC with the exceptions noted in Appendix A.

PROGRAM MENUS

One of the primary considerations in the development of these programs was that they be easy to learn and to use. As a result, a menu format is used in each of the programs described in this report. This format gives the user a large degree of control over the program operation. For example, the user can easily enter data as well as change incorrect values. Also, the menu approach gives the user control over initiation of program calculations, so data can be entered or changed any number of times before calculations are executed. The paragraphs below present an overview of what a menu is and how it is used in the programs that follow.

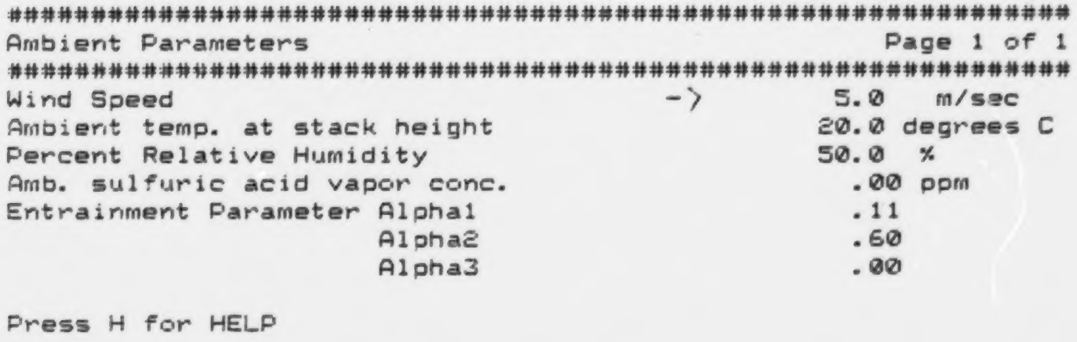
All of the programs begin with a Main Menu. The Main Menu is the "hub" of program operations. Figure 1-1 shows an example of a Main Menu from the data entry program for the Plume Opacity (PLUMIN) progam. This menu is displayed on the computer screen at the start of the PLUMIN program. The menu lists several items and includes a pointer next to one of the items. Each of the items in the Main Menu is a subroutine (or a set of subroutines) that can be called by the user. The pointer is next to the "current" item in the menu. The pointer tells the program what item will be called when the <ENTER> key is pressed. For example, in Figure 1-1, the current item is called "Input Ambient parameters." To call this subroutine the user presses the <ENTER> key. In this sense, the Main Menu is the hub of the program because all subroutines are called from and return to it. If a user wanted to specify the characteristic values for the plume (e.g. stack diameter and exit velocity), he would move the pointer to the second item in the Main Menu ("Input Plume parameters") by pressing the up or down arrow keys until the pointer was next to that item and then pressing <ENTER>. When finished with that subroutine, the program would return the user to the Main Menu. In this way the program user could execute any of the subroutines listed in the Main Menu in any order, as many times as necessary. It is this flexibility that is the true advantage of the menu approach to programming.

The Main Menu described above always consists of a number of subroutines that can be called by the user. Another type of menu is the "data menu," which allows the user to enter data into a program. There are two types of data menus. The first type is used in the baghouse and opacity programs. The example shown in Figure 1-2 is displayed when the user calls the "Input



```
****************************************************************
                      RTI Plume model
****************************************************************
              ->    Input Ambient parameters
                    Input Plume parameters
                    Save Input Parameters in a File
                    Read Input Parameters from a File
                    Exit


                    Press H for HELP
```

Figure 1-1. The Main Menu for the PLUMIN data entry program.



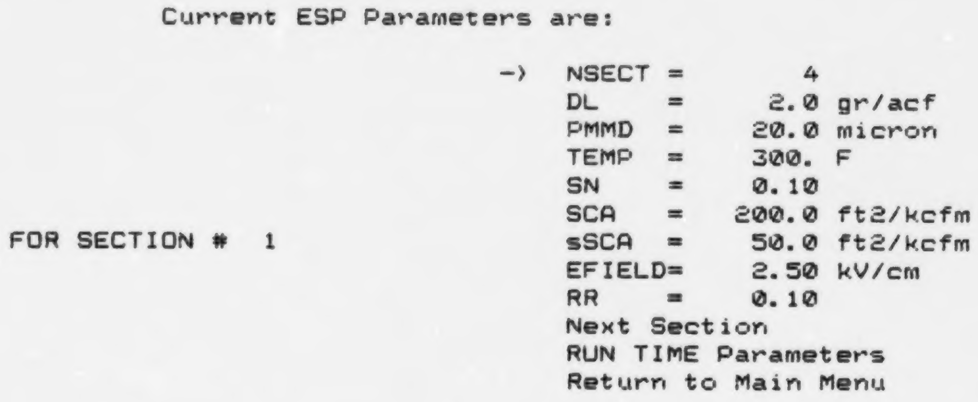
```
################################################################
Ambient Parameters                                  Page 1 of 1
################################################################
Wind Speed                             ->        5.0   m/sec
Ambient temp. at stack height                   20.0 degrees C
Percent Relative Humidity                       50.0  %
Amb. sulfuric acid vapor conc.                    .00 ppm
Entrainment Parameter Alpha1                      .11
                      Alpha2                      .60
                      Alpha3                      .00

Press H for HELP
```

Figure 1-2. A data entry menu from the PLUMIN program.

Ambient parameters" subroutine from the Main Menu listed in Figure 1-1. In this type of menu the description of an item is in the left column on the screen and the current value is listed in the right column. For example, in Figure 1-2, the "Wind Speed" is currently 5.0 m/sec, "Ambient temp" is 20.0 degrees C, and so forth. To change any of the currently listed values, press the up or down arrow keys on the keyboard to position the pointer next to the item to be changed and then type in the new value. For example, to set the ambient temperature to 25.0 °C, move the pointer to the second item in the menu, type in 25.0 and press <ENTER>. The menu will automatically be rewritten with the new value shown. The pointer can be moved to any position in the menu and any value can be changed. When all of the values have been entered, return to the Main Menu by pressing the capitol <R> key. Note that the method used to return to the Main Menu varies between programs but is specified in each.

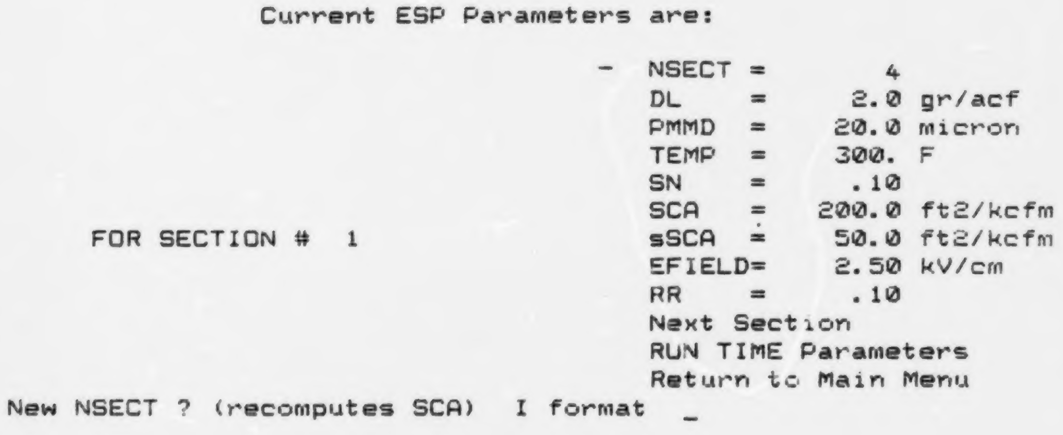
The second type of data menu is shown in Figure 1-3. In this type of menu the parameter name is listed along with its current value. The pointer is to the left of the list of parameter names. To change one of the values listed in the menu, position the pointer next to the parameter name and press the <ENTER> key. The program will print a message along the bottom of the screen asking for the new value and specifying the FORTRAN format for the number. Type in the new value and press <ENTER>. The program will rewrite the screen with the new value and wait for another command. For example, to change the number of sections in this menu, position the pointer next to the line labelled "NSECT" and press <ENTER>. The program will respond with a question along the bottom line of the screen as shown in Figure 1-4. Type in a new value (e.g., 6) and press <ENTER>. The screen will be redrawn with the new value shown. It is not essential that the user be familiar with FORTRAN format statements to use this program. A simple rule of thumb should suffice: if the requested format contains a decimal point, put a decimal point in the entry. If the requested format does not contain a decimal point, then do not put one in. In this type of menu the last item is always the "Return to Main Menu" option. Move the arrow to point to this option and press <ENTER> to return to the Main Menu.

Both of these menu types are easy to use and you will become familiar with them with a little practice.



```
                 Current ESP Parameters are:

                                        ->  NSECT =       4
                                            DL    =     2.0 gr/acf
                                            PMMD  =    20.0 micron
                                            TEMP  =    300. F
                                            SN    =    0.10
                                            SCA   =   200.0 ft2/kcfm
     FOR SECTION #  1                       sSCA  =    50.0 ft2/kcfm
                                            EFIELD=    2.50 kV/cm
                                            RR    =    0.10
                                            Next Section
                                            RUN TIME Parameters
                                            Return to Main Menu
```

Figure 1-3. A data entry menu from the dynamic ESP model.



```
                    Current ESP Parameters are:

                                           -  NSECT =       4
                                              DL    =     2.0 gr/acf
                                              PMMD  =    20.0 micron
                                              TEMP  =    300. F
                                              SN    =     .10
                                              SCA   =   200.0 ft2/kcfm
        FOR SECTION #  1                      sSCA  =    50.0 ft2/kcfm
                                              EFIELD=    2.50 kV/cm
                                              RR    =     .10
                                              Next Section
                                              RUN TIME Parameters
                                              Return to Main Menu
    New NSECT ? (recomputes SCA)  I format  _
```

Figure 1-4. Request for data in dynamic ESP model.

CHAPTER 2

ESP SECTION FAILURE MODEL

Program Author: Dr. P. A. Lawless

REVISION 1.0

ABSTRACT

This program calculates the bare penetration of an electrostatic precipitator (ESP) according to a simplified collection process that takes into account sneakage and rapping reentrainment. The user is then allowed to input a fraction of the ESP that is in failure (in one or more electrical sections) or to input a velocity distribution function. The program then recalculates the ESP penetration for each adverse condition and combines the resulting emissions for a common output.

USER BENEFITS

- The ESP sectional model allows a quick look at the factors influencing ESP performance: electric field, particle size, specific collection area (SCA), and losses.
- The section failure model allows a quick look at the relative significance of a section failure as guidance for more detailed modeling.
- The velocity maldistribution model allows the user to understand the relative importance of correcting gas flow problems and the relative improvement that might be expected.
- By using the program to screen several cases in advance, the user can save considerable time in using more complex ESP models.
- This failure model addresses ESP performance in a new and logical way. It will be a useful training tool for those not familiar with other more complicated ESP computer models.

HARDWARE/SOFTWARE REQUIREMENTS

This program runs on a TRS-80 Model I, III, or 4 microcomputer with the NEWDOS/80 Version 2 (Apparat, Inc., 4401 S. Tamarac Parkway, Denver, CO) operating system. It requires 48K of RAM and at least one disk drive. It will accommodate any printer that has been set up to print a 132-column line. A printer is useful for program output, but it is possible to use the program without a printer, since all relevant data are displayed on the screen.

The program is written mainly in Microsoft FORTRAN (Microsoft Corporation, 13221 SE 26th Street, Belleview, WA), but references some subroutines written in ASSEMBLER. If any routines are to be modified, the Microsoft programs F80, M80, and L80 would be required.

Since the program uses machine- and ROM-dependent coding, this version will be expected to run only on the TRS-80 machines mentioned above with the NEWDOS operating system.

METHOD

The penetration of an ESP is calculated section by section, with the total penetration given by the product of section penetrations:

$$P_t = P_1 \cdot P_2 \cdot P_3 \cdot \ldots \quad . \qquad (2\text{-}1)$$

Each section penetration is assumed to be the sum of two terms, the uncollected fraction and the loss fraction:

$$P_n = LF + (1 - LF) \cdot P_c \ . \qquad (2\text{-}2)$$

The uncollected fraction is given by the Deutsch-Anderson (D-A) equation:

$$P_c = \exp(-w \cdot SCA) \qquad (2\text{-}3)$$

where w is the particle migration velocity and SCA is the specific collection area.

The loss fraction is given in terms of the sneakage (SN) and rapping reentrainment fractions (RR) by

$$LF = SN + RR - SN \cdot RR \ . \qquad (2\text{-}4)$$

These loss terms represent the fraction of the mass in the ESP section, which is either not collected (SN) or which is collected and then lost again (RR). The default values in the program (SN=RR=0.1) are those that give good results for most ESPs.

The SCA is the specific collection area for the electrical section being studied, not for the whole ESP.

The D-A migration velocity, w, is calculated for one particle size under the following assumptions:

- The minimum particle size encountered is 2 μm diameter.
- Field charging is assumed to account for the particle charge.
- The particle is assumed to reach the saturation charge characteristic of the electric field for the section it is in.
- The migration velocity is equal to the electric force divided by the viscous drag coefficient for that particle size.
- No Cunningham correction factor is used.

Under these assumptions, the migration velocity is given by:

$$w = d \cdot \varepsilon \cdot E^2/\eta \qquad (2\text{-}5)$$

where

d = mean particle diameter

ε = free space permittivity

E = electric field

η = the gas viscosity.

The mean particle diameter is assumed to change throughout the length of the ESP, although it is recalculated only at the outlet of each section. The diameter assumed at the inlet of the ESP is the mass median diameter (MMD) of the particle distribution. This is done for three reasons: it is often the only value known; it is usually large enough for all the assumptions to apply; and, under most ESP theories, all particles larger than the MMD will be collected at least as well as those at the MMD.

The particle diameter is always calculated as a penetration weighted average of several particle diameters. For the uncollected fraction in the section, the diameter of particles emerging from the section is a weighted average between the inlet MMD and 2 μm at the outlet:

$$\mathrm{MMD}_{out} = \mathrm{MMD}_{inlet} \cdot P_c + 2.0 \cdot (1 - P_c) \ . \qquad (2\text{-}6)$$

Thus, if a section collects poorly, the MMD at the outlet is close to the

MMD at the inlet; but if it collects well, the outlet MMD is about 2 μm. (The choice of 2 μm is based on studies of detailed models of ESP performance.)

For the fraction of particles that sneak through the section, the MMD at the outlet is the same as the MMD at the inlet. For the fraction of particles at the outlet due to rapping reentrainment, the MMD is fixed at 6 μm. These three characteristic sizes are combined at the outlet to calculate a single particle diameter for input to the next section:

$$MMD_{total} = (P_c \cdot MMD_{out} + SN \cdot MMD_{inlet} + RR \cdot 6.0)/(P_c + SN + RR). \qquad (2\text{-}7)$$

An approximation for the opacity calculation is that the particle diameter appropriate for opacity is the outlet MMD, but with a scattering cross section fixed for a smaller particle size. The opacity calculation is intended to indicate trends rather than accurate results.

The real value of this program is its ability to simulate the effects of abnormal operation. One type of abnormal operation results from velocity maldistribution. Ideally, the velocity in an ESP should be very uniform across the flow channel. In practice, there can be wide variations about the mean velocity. The effect of velocity maldistribution can be visualized by breaking the ESP into a set of parallel flow channels with different velocities. A large fraction of the flow through the ESP will travel at the mean velocity. A smaller portion of the flow will travel with slightly higher and lower velocities. Still less of the flow will travel at velocities widely different from the mean. If these parallel channels are considered independently, the collection efficiency of the path where the velocity is below the mean will be greater than the average value because the lower velocity yields a higher SCA for that path. Conversely, for the path where the velocity is greater than the mean, the SCA will be lower than the average and hence the collection efficiency will be reduced. The overall SCA is the same as for the uniform velocity case, but the performance is degraded by the velocity maldistribution. When the variations in velocity become large then it is possible to have a path where there is essentially no flow, thereby reducing the overall SCA. For lack of experimental data, it is assumed that the variations in velocity follow a normal distribution. The microcomputer program allows the user to specify the mean velocity and the normalized standard deviation (the ratio of the standard deviation to the mean) to describe the maldistribution in an ESP.

Another type of abnormal operation arises from electrical and/or mechanical failures that result in a loss of collecting area. For example, consider the ESP sketched in Figure 2-1a. In this sketch (shown from the top, looking down on the precipitator), there are two independent flow channels with four electrical sections in each channel. If one of the electrical sections in flow channel 1 fails, the precipitator would appear as in Figure 2-1b. In this case, half of the ESP has no failures (flow channel 2) and half of the ESP has one section failure (flow channel 1). As a result of the section failure, the penetration of dust through channel 1 will be greater than the penetration through channel 2. Now consider the case where another section fails in channel 1. This is depicted in Figure 2-1c. In this case, half of the ESP has zero failures and half of the ESP has two failures. Here the penetration through channel 1 will be much greater than through channel 2 because the SCA in channel 1 is effectively half of that in channel 2. An important comparison can be drawn between the cases depicted in Figures 2-1c and 2-1d. In Figure 2-1d, the total number of failed sections is the same, but in this case the failed sections are distributed between the two flow channels. The overall penetration for this case (2-1d) will be lower than for case 2-1c, even though the amount of plate area out of service is the same. This is an important point: it is not sufficient to say how much plate area has failed; it is also necessary to say how that plate area is distributed among the independent flow channels of the ESP. This point will be further illustrated in the sample problem given below.

The preceding description is primarily intended as an overview of the model details. A more thorough description of the theoretical aspects of the model as well as the practical application of the program is included in Appendix B.

GENERAL RESTRICTIONS

Although the program will deliver results for almost any inputs, negative values should not be used for any of the input quantities, and zero values should not be used for quantities representing physical dimensions. The results are unpredictable in those cases. The model will probably exhibit large errors if the inlet MMD is 6 μm or less because the assumptions about particle diameter are based on experience with fly ash and the rapping

10

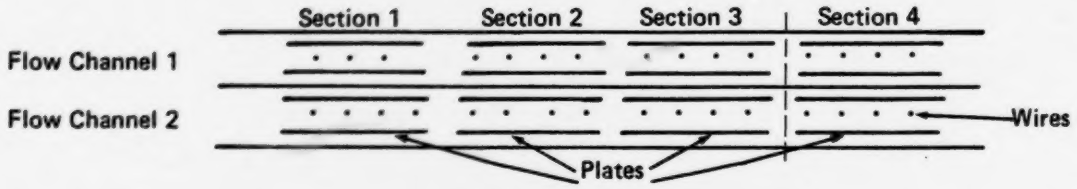


Figure 2-1a. A hypothetical ESP.

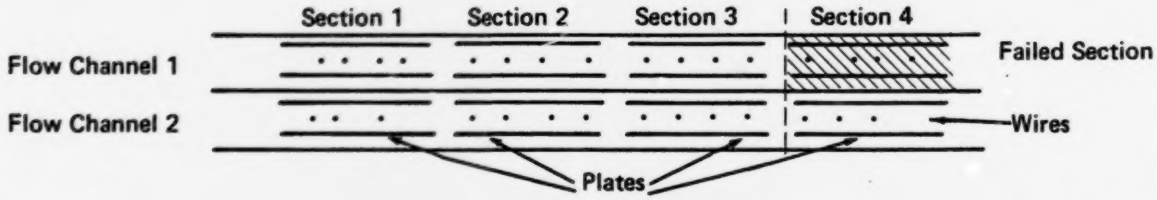


Figure 2-1b. ESP with one failed section.

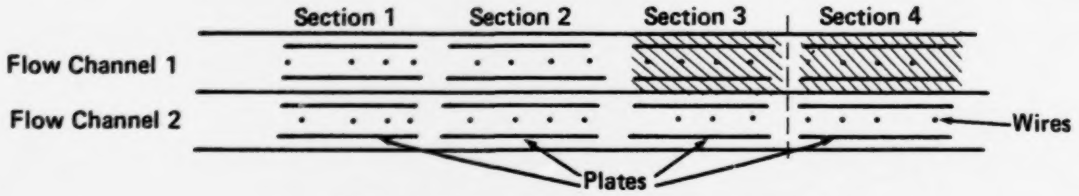


Figure 2-1c. Two failed sections in channel 1.

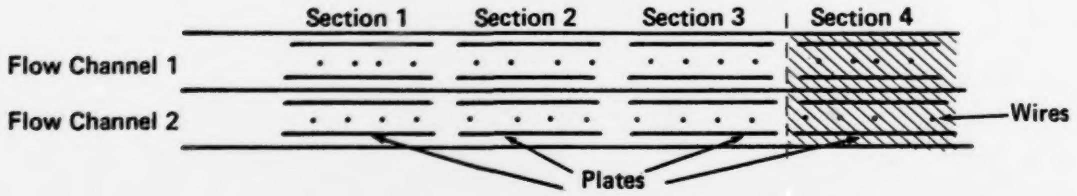


Figure 2-1d. Two failed sections; one in each channel.

reentrainment MMD is empirically determined on fly ash ESPs. If a smaller MMD is used at the inlet, the outlet MMD will be seen to grow to a larger size than the inlet.

There is no simple way to account for back-corona effects in this model. Some empirical measurements suggest that back corona reduces particle charge by a factor of 0.5 to 0.3; therefore, the penetration through a section could be changed by a similar factor, for example, by multiplying the SCA by the same factor or by multiplying the electric field by the square root of the factor (since the penetration depends quadratically on the field).

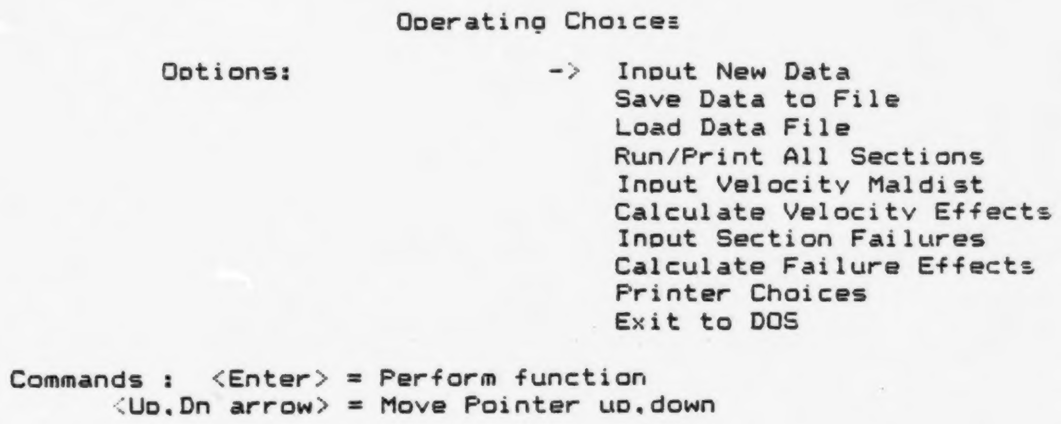
The best method of calculating electric field for conventional ESPs is to divide the corona electrode voltage by the electrode-plate distance and further divide the quotient by a factor of 1.5 to 1.75. Use the smaller divisor for closely spaced corona wires, or mast electrodes, and the larger divisor for widely spaced wire electrodes.

SAMPLE PROBLEM

Statement: Assume you have a precipitator with four electrical sections and an SCA of 300 ft^2/kacfm operating as a typical cold-side unit on fly ash. Calculate what the effects on emissions would be if one section went out of service in half of the flow channels in the ESP.

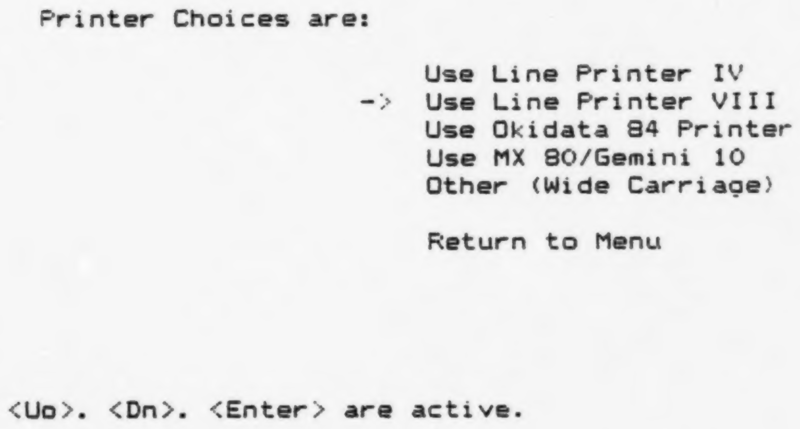
Results: This sample problem serves two purposes. First, the user can compare his/her results with the numbers listed below to verify that the program is operating properly. Second, the step-by-step instructions given below will help the novice learn how to use the program. Since this section is intended as a tutorial, the reader is encouraged to sit in front of the computer and try each step as it is described.

When the program begins execution the title of the program will appear on the screen along with a brief description of what the program does. After these messages, the Main Menu of operating choices will appear on the screen. This menu is shown in Figure 2-2. As mentioned in Chapter 1, this menu presents a list of subroutines that can be called to perform various functions. Near the bottom of the menu is the option, "Printer Choices." This subroutine should be called first so that the program will know what type of printer (if any) is connected to the computer. The "Printer



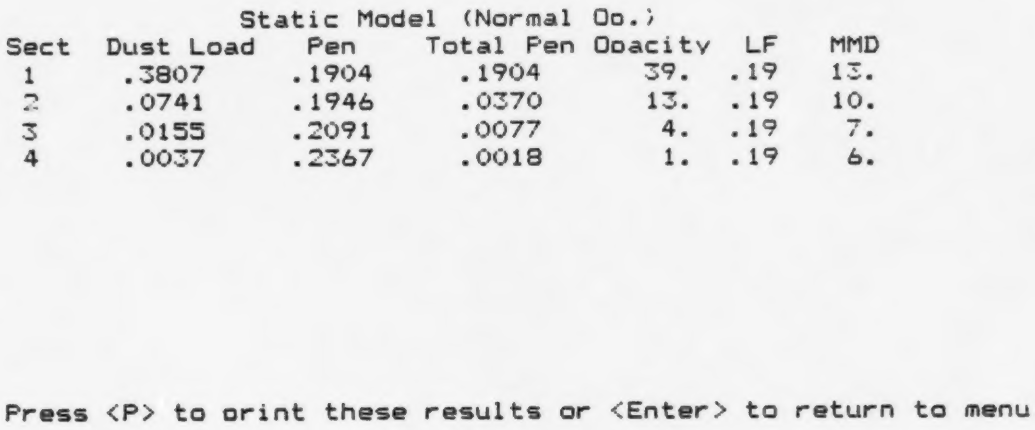
```
                    Operating Choices

      Options:                 ->  Input New Data
                                   Save Data to File
                                   Load Data File
                                   Run/Print All Sections
                                   Input Velocity Maldist
                                   Calculate Velocity Effects
                                   Input Section Failures
                                   Calculate Failure Effects
                                   Printer Choices
                                   Exit to DOS

Commands :  <Enter> = Perform function
      <Up,Dn arrow> = Move Pointer up,down
```

Figure 2-2. The Main Menu for the ESP section failure program.



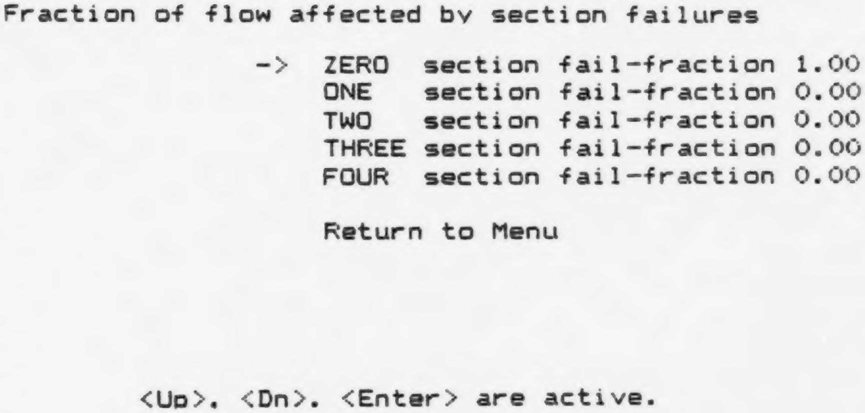
```
   Printer Choices are:

                            Use Line Printer IV
                        ->  Use Line Printer VIII
                            Use Okidata 84 Printer
                            Use MX 80/Gemini 10
                            Other (Wide Carriage)

                            Return to Menu



<Up>, <Dn>, <Enter> are active.
```

Figure 2-3. The printer choices menu.



```
                 Static Model (Normal Op.)
Sect   Dust Load    Pen      Total Pen Opacity  LF    MMD
 1      .3807      .1904      .1904       39.   .19    13.
 2      .0741      .1946      .0370       13.   .19    10.
 3      .0155      .2091      .0077        4.   .19     7.
 4      .0037      .2367      .0018        1.   .19     6.




Press <P> to print these results or <Enter> to return to menu
```

Figure 2-5. Results for the normal case.



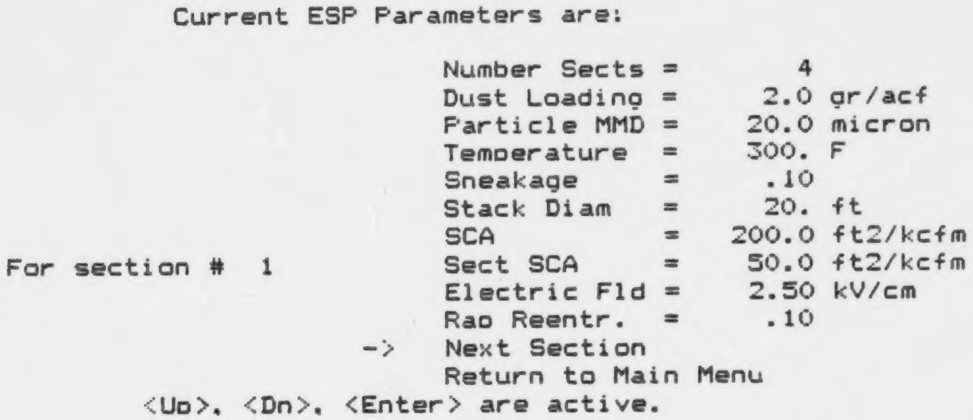
```
Fraction of flow affected by section failures

                    ->  ZERO  section fail-fraction 1.00
                        ONE   section fail-fraction 0.00
                        TWO   section fail-fraction 0.00
                        THREE section fail-fraction 0.00
                        FOUR  section fail-fraction 0.00

                        Return to Menu



          <Up>. <Dn>. <Enter> are active.
```

Figure 2-6. "Input Section Failures" menu.

Choices" menu is shown in Figure 2-3. To choose one of the printers included in the list, position the blinking arrow next to your selection and press <ENTER>. If your printer is not included in the list then select the "Other (Wide Carriage)" option. Note that this program will work properly with any printer but that on some of the small carriage printers there may be a problem of "wrap around." The program automatically returns to the Main Menu when you press <ENTER>.

Once the printer has been specified it is time to enter the data for the problem at hand. From the Main Menu choose the "Input New Data" option. The data menu shown in Figure 2-4 should appear on the screen. There are two parts to this menu. The first part consists of the first seven items in the menu, which pertain to the entire ESP and/or the ESP inlet conditions. The second part of the menu begins on the line that says, "For section # 1." This lower portion lists the values of sectional SCA, electric field, and rapping reentrainment factor that apply to section 1 of the ESP. If one of the first seven items in this menu is changed then that new value pertains to the entire ESP. If one of the next three parameters is changed, the new value applies only to the specified section. In Figure 2-4 the parameters listed are for section 1. To view the parameters for section 2, move the blinking arrow to the menu item, "Next Section" and press <ENTER>. The top portion of the menu will remain unchanged and the lower portion will show the parameters for ESP section 2. Subsequent calls to the "Next Section" command will show the parameters for the succeeding sections of the ESP. The program returns to section 1 after the last section is displayed.

The use of the data menu is fairly straightforward but there is one point that may cause some confusion. The SCA listed in the upper portion of the data menu is for the entire ESP. The sectional SCA listed below it is for the specified section. If you enter the overall SCA, the program recomputes all the sectional SCAs by dividing the overall SCA by the number of sections present. However, if you enter a sectional SCA, the overall SCA will be computed as the sum of the sectional SCAs. For example, in the sample problem, an SCA of 300 ft^2/kacfm is given. Enter this value into the data menu by positioning the arrow next to "SCA" and pressing <ENTER>. When the program prompts you for the new value, type <300>. Note that the sectional SCAs are recomputed (sectional SCA = 75 ft^2/kacfm) and shown in the



```
          Current ESP Parameters are:

                        Number Sects =       4
                        Dust Loading =     2.0 gr/acf
                        Particle MMD =    20.0 micron
                        Temoerature  =    300. F
                        Sneakage     =     .10
                        Stack Diam   =     20. ft
                        SCA          =   200.0 ft2/kcfm
 For section #  1       Sect SCA     =    50.0 ft2/kcfm
                        Electric Fld =    2.50 kV/cm
                        Rao Reentr.  =     .10
                   ->   Next Section
                        Return to Main Menu
        <Uo>, <Dn>, <Enter> are active.
```

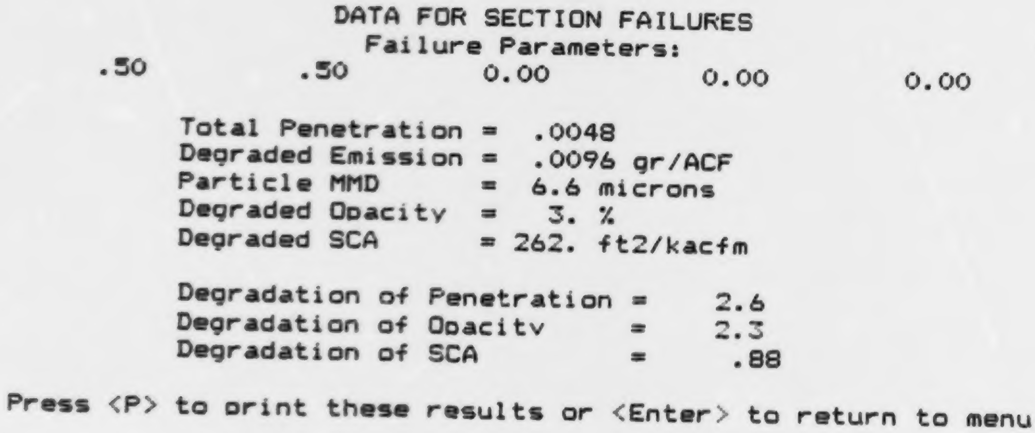
Figure 2-4. Data entry menu.

menu. Alternatively, you could have entered the sectional SCA as 75 ft^2/kacfm for each section and the total SCA would then be computed as 300 ft^2/kacfm.

Now that the SCA has been specified, return to the Main Menu and call the "Run/Print All Sections" option. This subroutine will compute the penetration, dust load, and opacity for the ESP. This step must be performed before examining the effects of abnormal operation because the program compares the normal and abnormal cases. The results shown in Figure 2-5 should appear on your screen. You may print these results on the printer by pressing the letter <P> on your keyboard.

Once the ESP results under normal operation have been computed, the effects of abnormal operation can be investigated. As mentioned earlier, the two types of abnormal operation that can be simulated in this model are velocity maldistribution and electrical/mechanical failure. For the sample problem, we are interested in an electrical failure. From the Main Menu, choose the "Input Section Failures" option. The menu shown in Figure 2-6 should appear on the screen. In the sample problem, we want to know the effect of losing one section in half of the ESP flow channels. This corresponds to the situation shown in Figure 2-1b, where one section of channel 1 failed. To specify this condition, set the blinking arrow to the "ZERO section fail-fraction" line and press <ENTER>. When the program asks for a number, type 0.5 and press <ENTER>. This means that in half (0.5) of the flow channels there are no section failures (flow channel 2 in Figure 2-1b). Now move the arrow to the line "ONE section fail-fraction" and press <ENTER>. Type in 0.5 and press <ENTER>. This signifies that half (0.5) of the ESP has lost one electrical section (flow channel 1 in Figure 2-1b). Note that the fractions listed along the right side of the display in Figure 2-6 should always add up to 1.0.

After the section failures have been specified, return to the Main Menu and call the "Calculate Failure Effects" option. The results shown in Figure 2-7 should appear on the screen. Near the top of this display the "Failure Parameters" are listed. This set of values is, from left to right, the fraction of the ESP channels with no section failures, the fraction with one section failure, the fraction of the ESP with two section failures, etc. Below these fractions are listed the penetration for the abnormal operation, emission level, dust MMD at the outlet, opacity, and new SCA (since some of



```
                    DATA FOR SECTION FAILURES
                       Failure Parameters:
   .50        .50        0.00          0.00          0.00

         Total Penetration =  .0048
         Degraded Emission =  .0096 gr/ACF
         Particle MMD      =  6.6 microns
         Degraded Opacity  =   3. %
         Degraded SCA      = 262. ft2/kacfm

         Degradation of Penetration =    2.6
         Degradation of Opacity     =    2.3
         Degradation of SCA         =     .88

Press <P> to print these results or <Enter> to return to menu
```

Figure 2-7. Results for the "Calculate Failure Effects" option.

the plate area has been removed from service). The lower group of lines shows the ratios of these numbers to the previously calculated values (i.e., normal operation) for easy comparison. Thus, losing an electrical section in half of the ESP causes the emissions to go up by a factor of 2.6, and the opacity goes up by a factor of 2.3 (because of a shift in average particle size). The SCA is reduced to 0.88 of its normal value (actually 0.875).

The results listed in Figure 2-7 provide the solution to the question posed in the sample problem. That is, when one section goes out of service in half of the ESP then the emission level increases by a factor of 2.6. The program operation steps performed in arriving at this result are

1. The "Printer Choices" option of the Main Menu was called to specify the type of printer that is connected to the computer.
2. The "Input New Data" option of the Main Menu was called to specify the overall SCA of the ESP. The SCA for each section was automatically recalculated as an equal fraction of the overall SCA.
3. The "Run/Print All Sections" option was called from the Main Menu to compute the results for an ESP operating normally. The results were listed as shown in Figure 2-5.
4. The "Input Section Failures" option was called to specify the number and distribution of section failures in the ESP.
5. The "Calculate Failure Effects" option was called to compute the results shown in Figure 2-7 for an ESP under abnormal operating conditions.

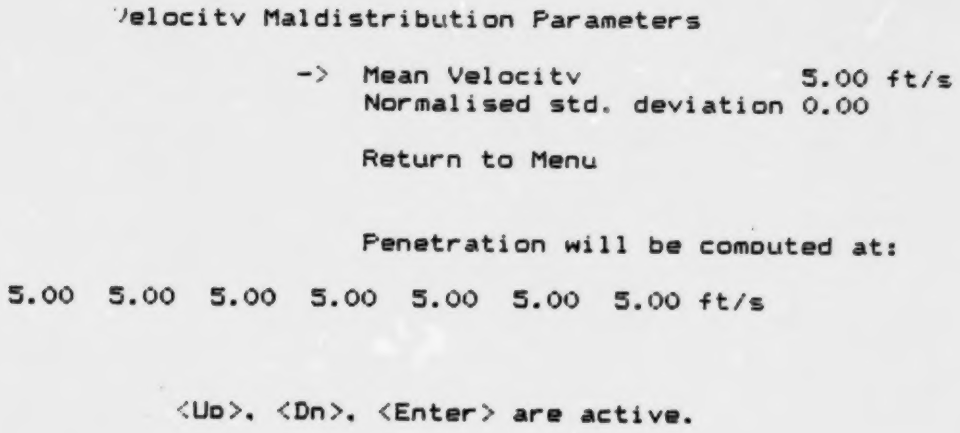
If at this point you want to see the effect of losing one section in the entire ESP, then you need only repeat steps 4 and 5 above. That is, once the printer type has been specified then step 1 does not have to be called again. As long as the overall ESP characteristics are the same, steps 2 and 3 do not have to be repeated. On the other hand, if you want to specify an ESP with a different design, you should return to step 2 to change the ESP parameters then go to step 3 to compute the normal operation results.

There are four options listed in the Main Menu (Figure 2-2) that have not yet been described. These are the data save and load options and the velocity maldistribution subroutines. The latter two subroutines are similar in operation to the section failure subroutines. The "Input Velocity Maldist" option of the Main Menu is used to specify the mean velocity through the ESP and the normalized standard deviation (NSD) for the velocity distribution.

This menu is shown in Figure 2-8. In the lower half of the screen is the message "Penetration will be computed at:" and a list of seven velocities at which the penetration will be calculated. When the NSD is zero, the velocity is uniform throughout the ESP, as shown in the list in Figure 2-8. When the NSD is greater than zero, the velocity is assumed to follow a normal distribution. For the case shown in Figure 2-9, the NSD is 0.25. The velocities that appear in the list in the lower half of the display describe the distribution function. Specifically, 90 percent of the flow through the ESP is between the velocities of 2.95 and 7.05 ft/sec, 60 percent of the flow is in the velocity range of 3.98 and 6.03 ft/sec, and 30 percent of the flow is in the velocity range 4.49 to 5.51 ft/sec.

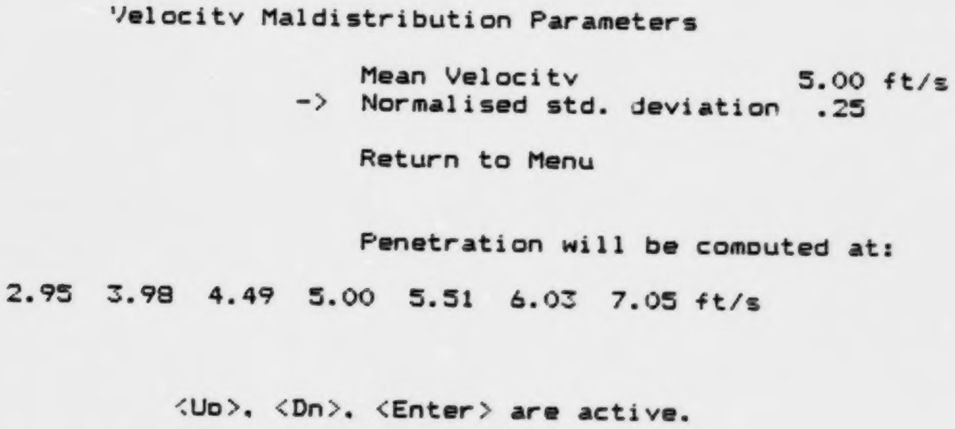
When the mean velocity and NSD have been specified, return to the Main Menu to calculate the effects of this maldistribution. From the Main Menu, call the "Calculate Velocity Effects" subroutine. The program will begin calculating the penetration and opacity at each of the seven velocities listed in Figure 2-9 and will flash the results on the screen briefly. After the individual results are computed, the program combines these results to give the penetration, emission level, particle size, opacity and SCA as shown in Figure 2-10. The velocity distribution function is listed at the top of the screen. The mean velocity is always in the middle of the velocity distribution list and the NSD is printed just below it. A hard copy of the results is available by pressing the <P> key. This printout is shown in Figure 2-11. The printout includes a comparison between the normal and abnormal operation results for Penetration, Opacity, and SCA.

The remaining two options in the Main Menu are the "Save Data to File" and "Load Data File" functions. These options are useful when one particular ESP configuration is to be used several times. Rather than typing in the input data every time the program is run, the values can be entered once and saved in a data file on a floppy diskette using the "Save ..." function. Then, at some point in the future when that same ESP configuration is needed, the "Load..." function can be used to recall the data from the disk file. Figure 2-12 shows the message that appears on the screen when the "Save..." function is called. A similar message appears for the "Load..." function. Refer to your DOS manual for rules on creating valid file names.



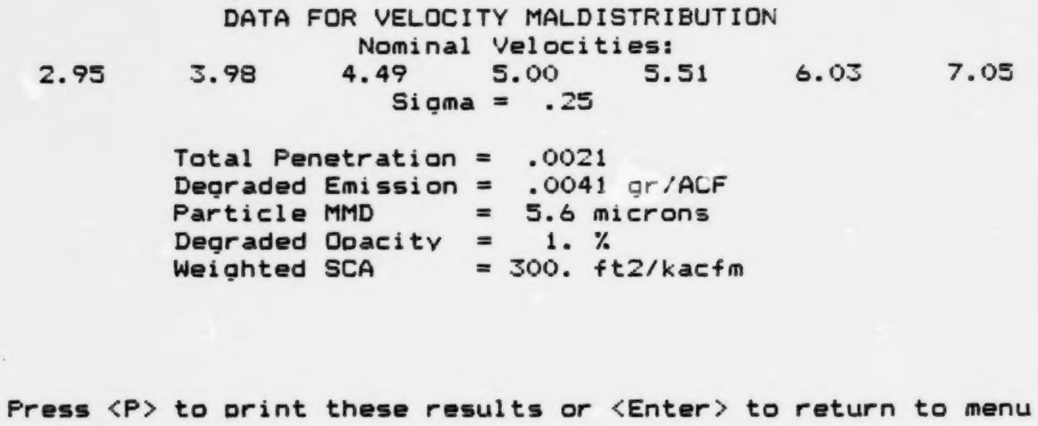
```
Velocitv Maldistribution Parameters

              ->  Mean Velocitv              5.00 ft/s
                  Normalised std. deviation  0.00

                  Return to Menu


                  Penetration will be comouted at:

5.00   5.00   5.00   5.00   5.00   5.00   5.00 ft/s


        <Uo>, <Dn>, <Enter> are active.
```

Figure 2-8. Velocity maldistribution input menu.



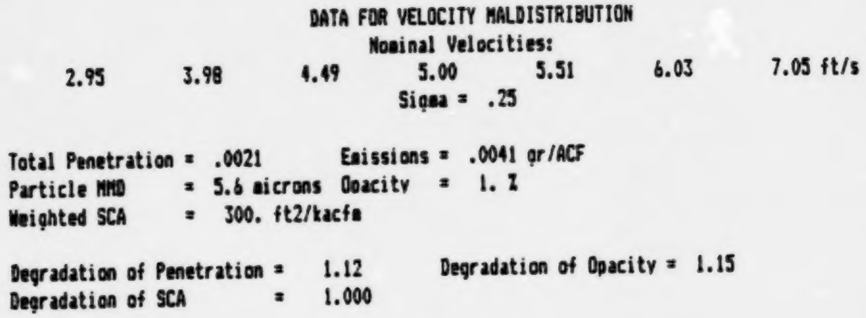
```
Velocitv Maldistribution Parameters

                  Mean Velocitv              5.00 ft/s
              ->  Normalised std. deviation   .25

                  Return to Menu


                  Penetration will be comouted at:

2.95   3.98   4.49   5.00   5.51   6.03   7.05 ft/s


        <Uo>, <Dn>, <Enter> are active.
```

Figure 2-9. Results of change in NSD in velocity maldistribution menu.



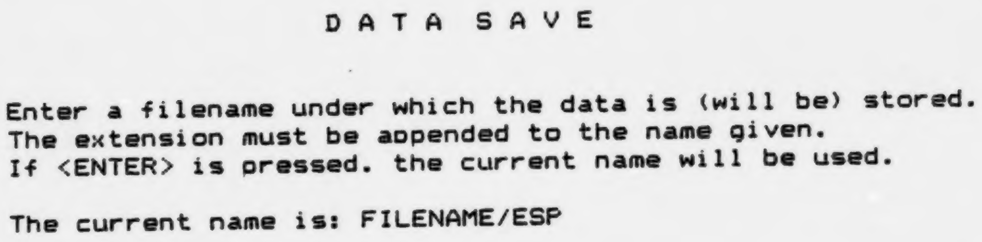
```
                 DATA FOR VELOCITY MALDISTRIBUTION
                        Nominal Velocities:
   2.95      3.98      4.49      5.00      5.51      6.03      7.05
                          Sigma =  .25

          Total Penetration =  .0021
          Degraded Emission =  .0041 gr/ACF
          Particle MMD      =  5.6 microns
          Degraded Opacity  =   1. %
          Weighted SCA      = 300. ft2/kacfm


Press <P> to print these results or <Enter> to return to menu
```

Figure 2-10. Computer screen results for "Calculate Velocity Effects".



```
                      DATA FOR VELOCITY MALDISTRIBUTION
                            Nominal Velocities:
   2.95      3.98      4.49      5.00      5.51      6.03      7.05 ft/s
                              Sigma =  .25

Total Penetration =  .0021       Emissions =  .0041 gr/ACF
Particle MMD      =  5.6 microns Opacity   =   1. %
Weighted SCA      =   300. ft2/kacfm

Degradation of Penetration =    1.12       Degradation of Opacity =  1.15
Degradation of SCA         =    1.000
```

Figure 2-11. Printout of velocity maldistribution effects.



```
                  D A T A   S A V E

Enter a filename under which the data is (will be) stored.
The extension must be appended to the name given.
If <ENTER> is pressed. the current name will be used.

The current name is: FILENAME/ESP
```

Figure 2-12. Screen response to "Save Data to File" option.

In summary, the program gives the user the ability to model normal ESP operation and abnormal conditions brought on by electrical failures and velocity maldistributions. The program allows the user to specify the important design parameters for the ESP ("Input New Data") and estimate the penetration under normal operating conditions ("Run/Print All Sections"). When the normal operation has been modelled, the user can specify a number of failed sections for any fraction of the ESP ("Input Section Failures") and compute the effect on performance ("Calculate Failure Effects"). Likewise, the user can enter velocity distribution parameters ("Input Velocity Maldist") and compute its effect on performance ("Calculate Velocity Effect"). For the sake of convenience, the program allows the user to save and load the ESP design parameters in data files and to specify the type of printer that is being used with the microcomputer.

```
      BLOCK DATA ESPDAT
C
C THIS PROGRAM SETS UP DEFAULT VALUES FOR THE ESP SECTION
C FAILURE MODEL.
C
      REAL MRAP,MMD
      INTEGER PSEL
      LOGICAL B3,B4,B5,B6,B,A,FN
C
      COMMON /EPPAR/NSECT,DL,PMMD,TEMP,SN,SCA,STCKDM
      COMMON /SECTL/SSCA(10),EFLD(10),RR(10)
      COMMON /FAIL/AX(6)
      COMMON /MET/TDK,SCAM(10),E(10),EPSVIS,OMSN
      COMMON /PRT/PSEL
      COMMON /SCPR/B3,B4,B5,B6,B(121),A(9)
      COMMON /OUTDAT/PENT(10),EMM(10),MMD(10),ISECT
      COMMON /DEGRES/XDL,XOP,XSC,OUTPEN,OUTDL,OUTOP,OUTSCA, OUTMMD
      COMMON /FVEL/VX0,VX(7),VSIG,VS(7)
      COMMON /FILDAT/FN(13)
C
      DATA NSECT,DL,PMMD,TEMP,SN,SCA/4,2.0,20.0,300.,
     * 0.1,200.0/
      DATA SSCA,EFLD,RR/10*50.0,10*2.5,10*0.1/
      DATA STCKDM/20./PSEL/2/
      DATA AX/1.0,0.,0.,0.,0.,0./
      DATA VS/7*5.0/VSIG/0.0/VX0/1.0/VX/7*1.0/
C
      DATA B3,B4,B5,B6,B,A/X'1B',X'0A',
     * X'14',X'1E',121*'+',9*'*'/
C
      END
```

ESP Section Failure Model Listing

```
      PROGRAM ESPDRV
C
C THIS PROGRAM DISPLAYS THE MENUS FOR ALL THE DATA OPTIONS.
C THE BLOCK DATA PROGRAM "ESPDAT" CONTAINS THE DEFAULT VALUES.
C
       INTEGER POINTR,COLUMN,ELMNTS
       EXTERNAL ESPDAT
C
C PUT UP A SCREEN MESSAGE AND WAIT FOR REPLY
C
       CALL SCRNMS
C
       POINTR = 1
C
1100   CONTINUE
C
C CLEAR SCREEN
C
       CALL CLS
C
       ELMNTS = 10
       COLUMN = 33
C
       WRITE(3,1)
  1    FORMAT('+',23X,'Operating Choices'
     * //'+',2X,'        Options:      ',14X,'Input New Data'
     * /'+',36X,'Save Data to File'
     * /'+',36X,'Load Data File'
     * /'+',36X,'Run/Print All Sections'
     * /'+',36X,'Input Velocity Maldist'
     * /'+',36X,'Calculate Velocity Effects'
     * /'+',36X,'Input Section Failures'
     * /'+',36X,'Calculate Failure Effects'
     * /'+',36X,'Printer Choices'
     * /'+',36X,'Exit to DOS'
     * //'+Commands :  <Enter> = Perform function'
     * /'+       <Up,Dn arrow> = Move Pointer up,down')
C
       CALL MENU(ELMNTS,COLUMN,POINTR)
C
       IF(POINTR.EQ.1) CALL DATAIN
C
       IF(POINTR.EQ.2) CALL SAVE
C
       IF(POINTR.EQ.3) CALL LOAD
C
       IF(POINTR.EQ.4) CALL RUNSTT(0)
       IF(POINTR.EQ.4) CALL PRTSCR
C
       IF(POINTR.EQ.5) CALL VELINP
C
       IF(POINTR.EQ.6) CALL VELCLC
C
       IF(POINTR.EQ.7) CALL SECFLR
C
```

```
       IF(POINTR.EQ.8) CALL CLCFLR
C
       IF(POINTR.EQ.9) CALL PRTOPT
C
       IF(POINTR.NE.10) GO TO 1100
C
       CALL CLS
       STOP
C
       END
```

ESP Section Failure Model Listing

```
      SUBROUTINE DATAIN
C
       INTEGER COLUMN,POINTR,ELMNTS
       COMMON /EPPAR/NSECT,DL,PMMD,TEMP,SN,SCA,STCKDM
       COMMON /SECTL/SSCA(10),EFLD(10),RR(10)
C
 700   ISECT = 1
C
C CLEAR THE SCREEN.
C
      POINTR = 11
      ELMNTS = 12
      COLUMN = 27
C
 300  CALL CLS
C
      WRITE(1,3) NSECT,DL,PMMD,TEMP,SN,STCKDM,SCA,ISECT,
     * SSCA(ISECT),EFLD(ISECT),RR(ISECT)
   3  FORMAT('+',15X,'Current ESP Parameters are:'//
     * '+',31X,'Number Sects =',I8/
     * '+',31X,'Dust Loading =',F8.1,' gr/acf'/
     * '+',31X,'Particle MMD =',F8.1,' micron'/
     * '+',31X,'Temperature  =',F8.0,' F'/
     * '+',31X,'Sneakage     =',F8.2/
     * '+',31X,'Stack Diam   =',F8.0,' ft'/
     * '+',31X,'SCA          =',F8.1,' ft2/kcfm'/
     * '+',5X,'For section # ',I2,10X,'Sect SCA     =',F8.1,
     *     ' ft2/kcfm'/
     * '+',31X,'Electric Fld =',F8.2,' kV/cm'/
     * '+',31X,'Rap Reentr.  =',F8.2/
     * '+',31X,'Next Section'/
     * '+',31X,'Return to Main Menu')
C
      CALL UPDN
C
      TPOINT = POINTR
C
      CALL MENU(ELMNTS,COLUMN,POINTR)
C
C CLEAR THE BOTTOM LINE OF SCREEN.
C
      CALL CBL
C
      GO TO(1100,1300,1500,1700,1900,2100,2300,2500,2700,
     * 2900,3100,3300),POINTR
1100  WRITE(1,5)
   5  FORMAT('+','New NSECT ? (recomputes SCA)  I format  ')
      READ(1,6) NSECT
   6  FORMAT(I2)
      IF(NSECT.GT.10) NSECT = 10
      IF(NSECT.LE.0.0) NSECT = 1
      ISECT = MIN0(ISECT,NSECT)
      SCA = 0.0
C
```

```
        DO 600 J = 1,NSECT
  600   SCA = SCA + SSCA(J)
        GO TO 300
C
 1300   WRITE(1,7)
    7   FORMAT('+New value for grain loading? F format  ')
        READ(1,8) DL
    8   FORMAT(F5.0)
        GO TO 300
C
 1500   WRITE(1,9)
    9   FORMAT('+New value for particle MMD ?  F format  ')
        READ(1,8) PMMD
        GO TO 300
C
 1700   WRITE(1,11)
   11   FORMAT('+New value for TEMPERATURE ? F format  ')
        READ(1,8) TEMP
        GO TO 300
C
 1900   WRITE(1,13)
   13   FORMAT('+New value for SNEAKAGE FACTOR ?  F format  ')
        READ(1,8) SN
        GO TO 300
C
 2100   WRITE(1,113)
  113   FORMAT('+New STACK DIAMETER (ft) ?  F format  ')
        READ(1,8) STCKDM
        GO TO 300
C
 2300   WRITE(1,17)
   17   FORMAT('+New SCA ? (recomputes sSCA) F format (0.0 ABRTS) ')
        READ(1,8) TSCA
        IF(TSCA.EQ.0.0) GO TO 300
        SCA = TSCA
C
        DO 200 J = 1,10
  200   SSCA(J) = TSCA/NSECT
        GO TO 300
 2500   WRITE(1,19)
   19   FORMAT('+New sSCA ? (recomputes SCA) F format (0.0 ABORTS) ')
        READ(1,8) TSCA
        IF(TSCA.EQ.0.0) GO TO 300
        SSCA(ISECT) = TSCA
        SCA = 0.0
C
        DO 400 J = 1,NSECT
  400   SCA = SCA + SSCA(J)
        GO TO 300
C
 2700   WRITE(1,21)
   21   FORMAT('+New EFIELD value ? (kV/cm) F format  ')
        READ(1,8) EFLD(ISECT)
        GO TO 300
C
```

ESP Section Failure Model Listing

```
2900    WRITE(1,15)
  15    FORMAT('+New value for RAPPING ENTRAINMENT ? F format  ')
        READ(1,8) RR(ISECT)
        GO TO 300
C
3100    ISECT = ISECT + 1
        IF(ISECT.GT.NSECT) ISECT = 1
        GO TO 300
C
3300    RETURN
C
        END
```

ESP Section Failure Model Listing

```
        SUBROUTINE SAVE
C
C   PROGRAM TO SAVE THE ESP DATA BLOCK
C
        LOGICAL FO,FN,TXT(36)
        COMMON /FILDAT/FN(13)
        DATA TXT/'D','U','M','P',' ','F','I','L','N','A','M',
     *  'E','1','/','E','S','P',' ','5','2','0','0','H',' ',
     *  '5','2','9','1','H',' ','4','2','0','D','H',X'0D'/
C
C CLEAR THE SCREEN
C
        CALL CLS
C
        WRITE(1,2)
   2    FORMAT('+',19X,'D A T A  S A V E')
C
        WRITE(1,3) (FN(I),I=1,12)
   3    FORMAT(
     *  //'+Enter a filename under which the data is (will be) stored.'
     *  /'+The extension must be appended to the name given.'
     *  /'+If <ENTER> is pressed, the current name will be used.'
     *  //'+The current name is:',1X,12A1/)
C
        CALL GETNAM
C
C CHECK FOR EXISTENCE OF A FILE ALREADY
C
        CALL FCHECK(FO,FN)
C
        IF(FO.EQ..FALSE.) GO TO 100
C
        WRITE(1,4)
   4    FORMAT(//'+FILE ALREADY EXISTS.  Press <BREAK> to abort save,'/
     *  '+or any other key to continue.')
C
        IASC = INKEY(1)
        IF(IASC.EQ.1) RETURN
C
C LOAD THE FILNAME INTO THE "DUMP" ARRAY
C
 100    DO 200 I = 1,12
        TXT(I+5) = FN(I)
 200    CONTINUE
C
        CALL DOSCLL(TXT)
C
        WRITE(1,7) (TXT(I),I=6,17)
   7    FORMAT(//'+ Data is stored in ',12A1/
     *  '+ Hit any key to RETURN')
C
        IASC = INKEY(1)
C
        RETURN
        END
```

```
      SUBROUTINE LOAD
C
C   PROGRAM TO LOAD THE ESP DATA BLOCK
C
        LOGICAL FO,FN,TX(18),CR
        COMMON /FILDAT/FN(13)
        DATA TX/'L','O','A','D',' ','F','I','L','N','A','M',
     *  'E','1','/','E','S','P',X'0D'/CR/X'0D'/
C
C CLEAR THE SCREEN
C
        CALL CLS
C
        WRITE(1,2)
   2    FORMAT('+',19X,'D A T A  L O A D')
C
        WRITE(1,3) (FN(I),I=1,12)
   3    FORMAT(
     *  //'+Enter a filename under which the data is (will be) stored.'
     *  /'+The extension must be appended to the name given.'
     *  /'+If <ENTER> is pressed, the current name will be used.'
     *  //'+The current name is:',1X,12A1/)
C
        CALL GETNAM
C
        CALL FCHECK(FO,FN)
C
        IF(FO.EQ..TRUE.) GO TO 300
C
        WRITE(1,4)
   4    FORMAT('+',19X,'FILE DOES NOT EXIST'//
     *  '+',19X,'Press any key to continue.')
C
        IASC = INKEY(1)
C
        RETURN
C
 300    DO 400 I = 1,12
        TX(I+5) = FN(I)
        IF(FN(I).EQ.32) TX(I+5) = CR
 400    CONTINUE
C
        CALL DOSCLL(TX)
C
        WRITE(1,5) (TX(I),I=6,17)
   5    FORMAT(//'+ Data has been loaded from ',12A1/
     *  '+ Hit any key to RETURN')
C
        IASC = INKEY(1)
C
        RETURN
C
        END
```

ESP Section Failure Model Listing

```
      SUBROUTINE GETNAM
C
       LOGICAL C,D,FN,BLNK
       COMMON /FILDAT/ FN(13)
       DATA C,D,BLNK/X'1B',X'1E',' '/
       DATA FN/ 'F','I','L','E','N','A','M','E','/', 'E','S','P',' '/
C
       DO 200 I = 1,12
       IASC = INKEY(1)
C
C CHECK FOR <ENTER>
C
       IF(IASC.EQ.13) GO TO 100
C
       FN(I) = IASC
C
C CHECK FOR LEFT ARROW KEY TO BACKSPACE
C
       IF(IASC.EQ.8) I = I - 2
C
       NUMGCH = I
C
C "C" IS AN UP LINEFEED CHARACTER, "D" IS ERASE TO END OF LINE
C
       WRITE(1,10) C,(FN(J),J=1,NUMGCH),D
 10    FORMAT('+'A1,21X,13A1)
C
 200   CONTINUE
C
 100   IF(I.EQ.1) RETURN
C
C BLANK OUT THE REST OF THE FILE NAME.
C
       NUMGCH = NUMGCH + 1
C
       DO 400 I = NUMGCH,13
 400   FN(I) = BLNK
C
       RETURN
C
       END
```

ESP Section Failure Model Listing

```
      SUBROUTINE PRTOPT
C
       INTEGER COLUMN,POINTR,ELMNTS,PSEL
       COMMON /PRT/PSEL
C
 100   POINTR = PSEL
C
C CLEAR THE SCREEN
C
       CALL CLS
C
       WRITE(1,2)
   2   FORMAT('+',15X,'Printer Choices are:'//
     * '+',38X,'Use Line Printer IV'/
     * '+',38X,'Use Line Printer VIII'/
     * '+',38X,'Use Okidata 84 Printer'/
     * '+',38X,'Use MX 80/Gemini 10'/
     * '+',38X,'Other (Wide Carriage)'//
     * '+',38X,'Return to Menu')
C
       CALL UPDN
C
       ELMNTS = 7
       COLUMN = 35
C
       CALL MENU(ELMNTS,COLUMN,POINTR)
C
       IF(POINTR.LE.5) PSEL = POINTR
C
       RETURN
C
       END
      SUBROUTINE VELINP
C
      INTEGER COLUMN,POINTR,ELMNTS
      COMMON /FVEL/VX0,VX(7),VSIG,VS(7)
      COMMON /EPPAR/NSECT,DL,PMMD,TEMP,SN,SCA,STCKDM
C
      POINTR = 1
      ELMNTS = 4
      COLUMN = 21
C
 100  VMEAN = VS(4)
C
      CALL CLS
C
C IDENTIFY THE SUBROUTINE
C
      WRITE(1,2) VMEAN,VSIG,VS
   2  FORMAT('+',9X,'Velocity Maldistribution Parameters'//
     * '+',24X,'Mean Velocity            ',F5.2,' ft/s'/
     * '+',24X,'Normalised std. deviation',F5.2//
     * '+',24X,'Return to Menu'///
     * '+',24X,'Penetration will be computed at:'//
     * '+',2X,F5.2,1X,F5.2,1X,F5.2,1X,F5.2,1X,F5.2,1X,
     * F5.2,1X,F5.2,1X,'ft/s')
C
      CALL UPDN
C
      CALL MENU(ELMNTS,COLUMN,POINTR)
C
      CALL CBL
C
      IF(POINTR.EQ.4) RETURN
      IF(POINTR.EQ.3) GO TO 100
      IF(POINTR.EQ.2) GO TO 300
      WRITE(1,3)
   3  FORMAT('+Enter nominal mean velocity of gas. ')
      READ(1,10) VS(4)
  10  FORMAT(F6.0)
      GO TO 500
C
 300  WRITE(1,4)
   4  FORMAT('+Enter normalised std. deviation of velocity. ')
      READ(1,10) VSIG
      VSIG = AMAX1(0.0,VSIG)
C
 500  CALL VNORM
C
      GO TO 100
C
      END
      SUBROUTINE VNORM
C
      DIMENSION SG(7)
      COMMON /FVEL/VX0,VX(7),VSIG,VS(7)
      DATA SG/-1.64,-0.82,-0.41,0.0,0.41,0.82,1.64/
C
C CLEAR THE SCREEN
C
      CALL CLS
C
C IDENTIFY THE SUBROUTINE
C
      WRITE(1,2)
    2 FORMAT(///////'+',19X,'N O R M A L I S I N G')
C
      VM = VS(4)
      VMIN = 0.05*VM
      SIG = VSIG*VM
      VX0 = 0.0
C
      DO 200 I = 1,7
      VS(I) = AMAX1(VM*(1.0 + SG(I)*VSIG),VMIN)
      VX(I) = VS(I)*EXP(-SG(I)**2/2.0)
  200 VX0 = VX0 + VX(I)
C
      RETURN
C
      END
```

```
       SUBROUTINE SECFLR
C
        INTEGER COLUMN,POINTR,ELMNTS
        COMMON/FAIL/AX(6)
        COMMON /EPPAR/NSECT,DL,PMMD,TEMP,SN,SCA,STCKDM
C
C CLEAR THE SCREEN
C
        POINTR = 1
C
 100    CALL CLS
C
C IDENTIFY THE SUBROUTINE
C
        WRITE(1,2) (AX(I),I = 1,5)
   2    FORMAT('+',5X,'Fraction of flow affected by section failures'//
     *  '+',24X,'ZERO  section fail-fraction',F5.2/
     *  '+',24X,'ONE   section fail-fraction',F5.2/
     *  '+',24X,'TWO   section fail-fraction',F5.2/
     *  '+',24X,'THREE section fail-fraction',F5.2/
     *  '+',24X,'FOUR  section fail-fraction',F5.2//
     *  '+',24X,'Return to Menu')
C
        CALL UPDN
C
        ELMNTS = 7
        COLUMN = 21
C
        CALL MENU(ELMNTS,COLUMN,POINTR)
C
        IF(POINTR.GT.5) GO TO 500
C
        IF(POINTR.GT.NSECT) GO TO 300
        IP = POINTR - 1
C
        CALL CBL
C
        WRITE(1,3) IP
   3    FORMAT('+Enter fraction of ESP with ',I1,' section(s) out: ')
        READ(1,10) AX(POINTR)
  10    FORMAT(F6.0)
        GO TO 100
C
 300    AX(POINTR) = 0.
        GO TO 100
C
 500    AX(6) = AX(1) + AX(2) + AX(3) + AX(4) + AX(5)
        IF(AX(6).NE.1.0) CALL NORM
        RETURN
C
        END
```

ESP Section Failure Model Listing

```
          SUBROUTINE NORM
C
          COMMON/FAIL/AX(6)
C
C CLEAR THE SCREEN
C
          CALL CLS
C
C IDENTIFY THE SUBROUTINE
C
          WRITE(1,2)
    2     FORMAT(///////'+',19X,'N O R M A L I S I N G')
C
          DO 200 I = 1,5
  200     AX(I) = AX(I)/AX(6)
C
          RETURN
C
          END


         SUBROUTINE SCRNMS
C
C THIS ROUTINE WILL DISPLAY TWO SCREEN MESSAGES.
C
          LOGICAL A(14),ONE,TWO
          DATA A/'L','O','A','D',' ','M','E','S','S','A',
     *    'G','E','1',X'0D'/ONE/'1'/TWO/'2'/
C
          A(13) = ONE
          CALL DOSCLL(A)
C
          IASC = INKEY(1)
          IF(IASC.EQ.1) RETURN
C
          A(13) = TWO
          CALL DOSCLL(A)
C
          IASC = INKEY(1)
C
          RETURN
          END
```

```
      SUBROUTINE UPDN
C
      CALL CBL
C
      WRITE(1,1)
    1 FORMAT('+',13X,'<Up>, <Dn>, <Enter> are active.')
C
      RETURN
C
      END


      SUBROUTINE VELCLC
C
      REAL MMD
      LOGICAL B3,B4,B5,B6,B,A
C
      COMMON /EPPAR/NSECT,DL,PMMD,TEMP,SN,SCA,STCKDM
      COMMON /SECTL/SSCA(10),EFLD(10),RR(10)
      COMMON /FAIL/AX(6)
      COMMON /MET/TDK,SCAM(10),E(10),EPSVIS,OMSN
      COMMON /SCPR/B3,B4,B5,B6,B(121),A(9)
      COMMON /OUTDAT/PENT(10),EMM(10),MMD(10),ISECT
      COMMON /DEGRES/XDL,XOP,XSC,OUTPEN,OUTDL,OUTOP,OUTSCA, OUTMMD
      COMMON /FVEL/VX0,VX(7),VSIG,VS(7)
C
      CALL CLS
C
      OUTPEN = 0.0
      OUTDL = 0.0
      OUTMMD = 0.0
      OUTSCA = 0.0
C
      DO 200 K = 1,7
      VD = VX(K)/VX0
C
      CALL RUNSTT(K)
C
      IF(K.EQ.4) PN = PENT(NSECT)
      IF(K.EQ.4) EMMN = EMM(NSECT)
      IF(K.EQ.4) PMDN = MMD(NSECT)
      OUTPEN = OUTPEN + PENT(NSECT)*VD
      OUTDL = OUTDL + EMM(NSECT)*VD
      OUTMMD = OUTMMD + MMD(NSECT)*VD
  200 OUTSCA = OUTSCA + SCA*VS(4)/VS(K)*VD
C
```

```
C
      WRITE(1,1)
    1 FORMAT('+',13X,'DATA FOR VELOCITY MALDISTRIBUTION')
C
      WRITE(1,2) (VS(I),I = 1,7),VSIG
    2 FORMAT(' ',21X,'Nominal Velocities:'/
     * '+',F6.2,3X,F6.2,3X,F6.2,3X,F6.2,3X,F6.2,3X,F6.2,
     * 3X,F6.2/
     * '+',23X,'Sigma = ',F4.2)
C
      OUTOP = CFUNC(OUTMMD,OUTDL)
C
      WRITE(1,7) OUTPEN,OUTDL,OUTMMD,OUTOP,OUTSCA
    7 FORMAT('0',10X,'Total Penetration = ',F6.4/
     * '+',10X,'Degraded Emission = ',F6.4,' gr/ACF'/
     * '+',10X,'Particle MMD      = ',6PF4.1,' microns'/
     * '+',10X,'Degraded Opacity  = ',0PF4.0,' %'/
     * '+',10X,'Weighted SCA      = ',0PF4.0,' ft2/kacfm')
C
      XDL = OUTDL/EMMN
      XOP = OUTOP/CFUNC(PMDN,EMMN)
      XSC = OUTSCA/SCA
C
  100 CALL PAKP(IASC)
C
      IF(IASC.EQ.13) RETURN
      IF(IASC.NE.80) GO TO 100
C
      CALL PRTVEL
C
      RETURN
C
      END
```

ESP Section Failure Model Listing

```
      SUBROUTINE RUNSTT(NV)
C
C THIS ROUTINE EVALUATES THE TIME AVERAGED PERFORMANCE OF
C THE ESP, USING THE D-A-R-N APPROACH.
C
      LOGICAL KS,A(64)
      REAL MMD,MRAP,LF
      COMMON /EPPAR/NSECT,DL,PMMD,TEMP,SN,SCA,STCKDM
      COMMON /SECTL/SSCA(10),EFLD(10),RR(10)
      COMMON /MET/TDK,SCAM(10),E(10),EPSVIS,OMSN
      COMMON /OUTDAT/PENT(10),EMM(10),MMD(10),ISECT
C
      NDEL = 0
      IF(NV.EQ.0) NDEL = 1
      IF(NV.EQ.0) NV = 4
C
C CONVRT PUTS QUANTITIES IN S-I UNITS AND INITIALISES.
C
      CALL CONVRT
C
      DINDL = DL
      TPEN = 1.0
C
C CLEAR THE SCREEN
C
      CALL CLS
C
      WRITE(1,3)
    3 FORMAT('+',14X,'Static Model (Normal Op.)'/
     * '+','Sect',2X,'Dust Load',3X,'Pen'4X,
     * 'Total Pen',1X,'Opacity',2X,'LF',3X,'MMD')
C
      DO 1000 KS = 1,NSECT
      ISECT = KS
      IF(KS.GT.1) MMD(KS) = MMD(KS-1)
C
C CALCULATE THE PENETRATION FOR THE SECTION.
C THE PENTRATION ROUTINE ALSO ALTERS THE MMD'S
C
      P = PEN(KS,NV)
C
C GET THE MASS RAPPED OFF.
C
      MRAP = (1.0 - P)*DINDL
      RAPDL = RR(KS)*MRAP
C
C GET THE MASS PENETRATING
C
      OUTDL = DINDL*P
      EMM(KS) = OUTDL + RAPDL
```

ESP Section Failure Model Listing

```
C
C CALCULATE THE WEIGHT RAP-PENETRATION MMD.
C
        MMD(KS) = (MMD(KS)*OUTDL + 6.0E-06*RAPDL)/EMM(KS)
C
C CALCULATE THE OVERALL PENETRATION.
C
        OUTDL = EMM(KS)
        P = OUTDL/DINDL
        DINDL = OUTDL
        TPEN = TPEN*P
        PENT(KS) = TPEN
C
C CALCULATE OPACITY FOR THIS STAGE
C
        OP = CFUNC(MMD(KS),EMM(KS))
C
C CALCULATE LOSS FACTOR FOR THIS STAGE
C
        LF = SN + RR(KS) - SN*RR(KS)
C
        WRITE(1,2) KS,OUTDL,P,TPEN,OP,LF,MMD(KS)
   2    FORMAT('+',I2,2X,F8.4,2X,F8.4,2X,F8.4,3X,F6.0,F5.2
     * ,6PF6.0)
C
1000    CONTINUE
C
        IF(NDEL.EQ.0) CALL DELAY
C
        RETURN
C
        END
```

ESP Section Failure Model Listing

```
      SUBROUTINE PRTSCR
C
       COMMON /EPPAR/NSECT,DL,PMMD,TEMP,SN,SCA,STCKDM
C
 100   CALL PAKP(IASC)
C
       IF(IASC.EQ.13) RETURN
C
       IF(IASC.NE.80) GO TO 100
C
       NLINES = NSECT + 4
C
C CHECK FOR PRINTER READY.
C
       CALL BUSY
C
C SET THE LINE PRINTER FOR 132 COLUMNS.
C
       CALL CONPRT
C
       CALL PRTHD
C
       CALL SCRPRT(NLINES)
C
       RETURN
C
       END
      FUNCTION PEN(N,NV)
C
       LOGICAL N
       REAL MMD
       COMMON /MET/TDK,SCAM(10),E(10),EPSVIS,OMSN
       COMMON /EPPAR/NSECT,DL,PMMD,TEMP,SN,SCA
       COMMON /OUTDAT/PENT(10),EMM(10),MMD(10),ISECT
       COMMON /FVEL/VX0,VX(7),VSIG,VS(7)
       DATA PI/3.14159/TWO/2.0E-06/
C
C THE PENETRATION IS CALCULATED FOR MMD AND USED TO FIGURE
C THE MEAN DIAMETER.  THIS CORRECTS A PROBLEM OCCURRING WHEN
C A LOW EFFICIENCY STAGE IS USED, viz., A SHIFT IN MMD.
C
C THE RATIO OF VELOCITIES IS USED TO ADJUST THE SECTIONAL SCA
C IN CALCULATING VELOCITY EFFECTS
C
       EF = EPSVIS*(E(N)*E(N))
       W = MMD(ISECT)*EF
       A = OMSN*EXP(-W*SCAM(N)*VS(4)/VS(NV))
       B = SN
C
       PEN = A + B
C
C CALCULATE A WEIGHTED AVERAGE BETWEEN THE INLET MMD AND 2
C MICRONS, DUE TO THE COLLECTION PROCESS.
C
       PMD = TWO*(1.0 - A) + MMD(ISECT)*A
C
C CALCULATE A NEW NON-RAPPING MMD LEAVING THE SECTION, PART DUE
C TO SNEAKAGE AND PART DUE TO COLLECTION.
C
       MMD(ISECT) = (A*PMD + B*MMD(ISECT))/PEN
C
       RETURN
C
       END
```

ESP Section Failure Model Listing

```
      SUBROUTINE CONVRT
C
       LOGICAL K
       REAL MMD
       COMMON /EPPAR/NSECT,DL,PMMD,TEMP,SN,SCA
       COMMON /SECTL/SSCA(10),EFLD(10),RR(10)
       COMMON /MET/TDK,SCAM(10),E(10),EPSVIS,OMSN
       COMMON /OUTDAT/PENT(10),EMM(10),MMD(10),ISECT
       DATA EPSO/8.845E-12/
C
       MMD(1) = PMMD*1.0E-06
       TDK = 5.0*(TEMP + 459.)/9.0
C
       OMSN = 1.0 - SN
C
       DO 200 K = 1,10
       SCAM(K) = SSCA(K)/5.079/OMSN
       E(K) = EFLD(K)*1.0E05
 200   CONTINUE
C
       VIS = 1.72E-05*(TDK/273.)**0.71
       EPSVIS = EPSO/VIS
C
       RETURN
C
       END
```

ESP Section Failure Model Listing

```
      SUBROUTINE PRTHD
C
       LOGICAL P1,P2
       COMMON /EPPAR/NSECT,DL,PMMD,TEMP,SN,SCA,STCKDM
       COMMON /SECTL/SSCA(10),EFLD(10),RR(10)
C
       WRITE(2,1)
    1  FORMAT('+',36X,'DATA FOR THE SECTIONAL MODEL')
C
       WRITE(2,2)
    2  FORMAT(' ',38X,'Precipitator Parameters:')
C
       WRITE(2,3) NSECT,SCA,SN,TEMP
    3  FORMAT('0',8X,'Num Sect = ',I2,2X,'Total SCA = ',
     * F5.0,' ft2/kACFM',2X,'Sneakage = ',F3.2,' /sect',
     * 2X,'Temp = ',F5.0,' Deg F.')
C
       WRITE(2,4) DL,PMMD,STCKDM
    4  FORMAT(' ',20X,'Dust Load = ',F4.1,' gr/acf',2X,
     * 'Part MMD = ',F4.1,2X,'Stack Diam = ',F3.0,' ft')
C
       WRITE(2,5) (SSCA(I),I=1,NSECT)
    5  FORMAT('0',15X,'Sectional SCA''s = ',10F7.1)
C
       WRITE(2,6) (EFLD(I),I=1,NSECT)
    6  FORMAT(' ',10X,'Sect Fields (kV/cm)  = ',10F7.2)
C
       WRITE(2,7) (RR(I),I=1,NSECT)
    7  FORMAT(' ',19X,'Rap Factors = ',10F7.2)
C
       RETURN
C
       END
      SUBROUTINE CLCFLR
C
       REAL MRAP,MMD
C
       COMMON /EPPAR/NSECT,DL,PMMD,TEMP,SN,SCA,STCKDM
       COMMON /SECTL/SSCA(10),EFLD(10),RR(10)
       COMMON /FAIL/AX(6)
       COMMON /MET/TDK,SCAM(10),E(10),EPSVIS,OMSN
       COMMON /OUTDAT/PENT(10),EMM(10),MMD(10),ISECT
       COMMON /DEGRES/XDL,XOP,XSC,OUTPEN,OUTDL,OUTOP,OUTSCA, OUTMMD
C
       CALL CLS
C
       WRITE(1,1)
    1  FORMAT('+',19X,'DATA FOR SECTION FAILURES')
C
       WRITE(1,2) (AX(I),I = 1,5)
    2  FORMAT(' ',21X,'Failure Parameters:'/
     * '+',2X,F6.2,6X,F6.2,6X,F6.2,7X,F6.2,6X,F6.2)
C
       JSECT = MAX0((NSECT - 4),1)
       OUTPEN = 0.0
       OUTDL = 0.0
       OUTMMD = 0.0
       OUTSCA = 0.0
C
       DO 200 I = JSECT,NSECT
       J = I - JSECT + 1
       IR = NSECT + JSECT - I
       OUTPEN = AX(J)*PENT(IR) + OUTPEN
       OUTDL = AX(J)*EMM(IR) + OUTDL
       OUTMMD = AX(J)*MMD(IR) + OUTMMD
  200  CONTINUE
C
       DO 400 I = 1,NSECT
       NST = NSECT + 1
       DO 400 J = 1,5
       IF(J.GT.(NST-I)) GO TO 400
       OUTSCA = OUTSCA + AX(J)*SSCA(I)
  400  CONTINUE
C
       OUTOP = CFUNC(OUTMMD,OUTDL)
C
       WRITE(1,7) OUTPEN,OUTDL,OUTMMD,OUTOP,OUTSCA
    7  FORMAT('0',10X,'Total Penetration = ',F6.4/
     * '+',10X,'Degraded Emission = ',F6.4,' gr/ACF'/
     * '+',10X,'Particle MMD      = ',6PF4.1,' microns'/
     * '+',10X,'Degraded Opacity  = ',0PF4.0,' %'/
     * '+',10X,'Degraded SCA      = ',0PF4.0,' ft2/kacfm')
C
       XDL = OUTDL/EMM(NSECT)
       XOP = OUTOP/CFUNC(MMD(NSECT),EMM(NSECT))
       XSC = OUTSCA/SCA
C
       WRITE(1,8) XDL,XOP,XSC
```

ESP Section Failure Model Listing

```
    8   FORMAT('0',10X,'Degradation of Penetration = ',F6.1/
     *  '+',10X,'Degradation of Opacity     = ',F6.1/
     *  '+',10X,'Degradation of SCA         = ',F7.2)
C
  100   CALL PAKP(IASC)
C
        IF(IASC.EQ.13) RETURN
        IF(IASC.NE.80) GO TO 100
C
        CALL PRTFLR
C
        RETURN
C
        END
```

```
      SUBROUTINE PRTVEL
C
       LOGICAL P1,P2
       REAL MMD
       LOGICAL B3,B4,B5,B6,B,A
C
       COMMON /EPPAR/NSECT,DL,PMMD,TEMP,SN,SCA,STCKDM
       COMMON /SECTL/SSCA(10),EFLD(10),RR(10)
       COMMON /FAIL/AX(6)
       COMMON /MET/TDK,SCAM(10),E(10),EPSVIS,OMSN
       COMMON /SCPR/B3,B4,B5,B6,B(121),A(9)
       COMMON /OUTDAT/PENT(10),EMM(10),MMD(10),ISECT
       COMMON /DEGRES/XDL,XOP,XSC,OUTPEN,OUTDL,OUTOP,OUTSCA, OUTMMD
       COMMON /FVEL/VX0,VX(7),VSIG,VS(7)
C
       CALL BUSY
       CALL CONPRT
C
       WRITE(2,1)
   1   FORMAT('+',33X,'DATA FOR VELOCITY MALDISTRIBUTION')
C
       WRITE(2,2) VS,VSIG
   2   FORMAT('+',39X,'Nominal Velocities:'/
     * '+'6X,F6.2,6X,F6.2,6X,F6.2,6X,F6.2,6X,F6.2,6X,
     * F6.2,6X,F6.2' ft/s'/
     * '+',42X,'Sigma = ',F4.2)
C
       WRITE(2,7) OUTPEN,OUTDL,OUTMMD,OUTOP,OUTSCA
   7   FORMAT('0',
     * 2X,'Total Penetration = ',F6.4,
     * 8X,'Emissions = ',F6.4,' gr/ACF'/'+',
     * 2X,'Particle MMD       = ',6PF4.1,' microns',
     * 2X,'Opacity   = ',0PF4.0,' %'/'+',
     * 2X,'Weighted SCA       = ',F6.0,' ft2/kacfm')
       WRITE(2,8) XDL,XOP,XSC
   8   FORMAT('0',2X,'Degradation of Penetration = ',F7.2,8X,
     * 'Degradation of Opacity = ',F5.2/
     * '+',2X,'Degradation of SCA         = ',F8.3/)
C
       RETURN
C
       END
```

ESP Section Failure Model Listing

```
      SUBROUTINE PRTFLR
C
       LOGICAL P1,P2
       REAL MMD
       LOGICAL B3,B4,B5,B6,B,A
C
       COMMON /EPPAR/NSECT,DL,PMMD,TEMP,SN,SCA,STCKDM
       COMMON /SECTL/SSCA(10),EFLD(10),RR(10)
       COMMON /FAIL/AX(6)
       COMMON /MET/TDK,SCAM(10),E(10),EPSVIS,OMSN
       COMMON /SCPR/B3,B4,B5,B6,B(121),A(9)
       COMMON /OUTDAT/PENT(10),EMM(10),MMD(10),ISECT
       COMMON /DEGRES/XDL,XOP,XSC,OUTPEN,OUTDL,OUTOP,OUTSCA, OUTMMD
C
       CALL BUSY
       CALL CONPRT
C
       NS = MIN0(5,NSECT+1)
C
       WRITE(2,1)
   1   FORMAT('+',37X,'DATA FOR SECTION FAILURES')
C
       WRITE(2,2) (AX(I),I = 1,NS)
   2   FORMAT('+',39X,'Failure Parameters:'/
     * '+',19X,'0 Sections',3X,'1 Section',3X,'2 Sections',2X,
     * '3 Sections',2X,'4 Sections'/
     * '+  Fraction:',9X,F6.2,6X,F6.2,6X,F6.2,6X,F6.2,6X, F6.2)
C
       WRITE(2,7) OUTPEN,OUTDL,OUTMMD,OUTOP,OUTSCA
   7   FORMAT('0',
     * 2X,'Total Penetration = ',F6.4,
     * 8X,'Emissions = ',F6.4,' gr/ACF'/'+',
     * 2X,'Particle MMD      = ',6PF4.1,' microns',
     * 2X,'Opacity   = ',0PF4.0,' %'/'+',
     * 2X,'Degraded SCA      = ',F6.0,' ft2/kacfm')
C
C
       WRITE(2,8) XDL,XOP,XSC
   8   FORMAT('0',2X,'Degradation of Penetration = ',F6.1,8X,
     * 'Degradation of Opacity = ',F4.1/
     * '+',2X,'Degradation of SCA         = ',F7.2/)
C
       RETURN
C
       END
```

ESP Section Failure Model Listing

```
      FUNCTION CFUNC(MMD,OUTDL)
C
      REAL KGPCM,MMD
      COMMON /EPPAR/NSECT,DL,PMMD,TEMP,SN,SCA,STCKDM
      DATA FORHPI,GRTOGM,HMD,QFAC/1256.6,0.00229, 0.7E-06,1.57/
C
C CALCULATE THE AVERAGE NUMBER OF PARTICLES PER CUBIC METER
C FROM THE OUTDL.
C
      KGPCM = OUTDL*GRTOGM
      GMD = MMD
C
C PARTICLE MASS IS 4*PI*RAD**3*DENSITY (2400 KG/M3).
C
      PARTMS = FORHPI*(GMD**3)
      PARTNM = KGPCM/PARTMS
C
C THE QEXT FACTOR IS 2.0 TIMES THE PROJECTED PARTICLE AREA.
C
      QEXT = QFAC*(GMD*GMD)
      BEXT = PARTNM*QEXT
      ABSOR = BEXT*STCKDM*0.305
      IF(ABSOR.LT.0.2) GO TO 100
C
      TRANS = EXP(-ABSOR)
      CFUNC = 100.*(1.0 - TRANS)
C
      RETURN
C
 100  CFUNC = 100.*ABSOR
C
      RETURN
C
      END
```

ESP Section Failure Model Listing

```
      SUBROUTINE SCRPRT(NLINES)
C
       LOGICAL A(64)
C
       CALL CONPRT
C
C PEEK THE SCREEN AND PRINT IT
C
       DO 200 I = 1,NLINES
       DO 400 J = 1,64
       IADR = X'3BFF' + 64*(I-1) + J
 400   A(J) = PEEK(IADR)
C
       WRITE(2,5) A
   5   FORMAT('+',22X,64A1)
C
 200   CONTINUE
C
       WRITE(2,6)
   6   FORMAT(' ')
C
       RETURN
C
       END


      SUBROUTINE CONPRT
C
       INTEGER PSEL
       LOGICAL B3,B4,B5,B6,P1,P2
       COMMON /PRT/PSEL
       COMMON /SCPR/B3,B4,B5,B6
       DATA P1,P2/X'1C',X'0F'/
C
C SET UP THE PRINTER FOR 132 COLUMNS
C
       IF(PSEL.LE.2) WRITE(2,3) B3,B5
       IF(PSEL.EQ.3) WRITE(2,3) P1
       IF(PSEL.EQ.4) WRITE(2,3) P2
   3   FORMAT('+',2A1)
C
       RETURN
C
       END
      SUBROUTINE PAKP(IASC)
C
       CALL CBL
C
       WRITE(1,4)
    4  FORMAT('+Press <P> to print these results or ',
     * '<Enter> to return to menu')
C
       IASC = INKEY(1).AND.95
C
       RETURN
C
       END
C
C
C
       SUBROUTINE DELAY
C
       DO 200 I = 1,20000
 200   CONTINUE
C
       RETURN
C
       END
```

```
        ******************** ASSEMBLY LANGUAGE ROUTINES *************************
****

                ENTRY INKEY,$INIT,MENU,DSTART,CLS,CBL,FCHECK
                ENTRY DOSCLL,BUSY
                EXTRN $EC,$CLSFL
;
; reserve space for all the variables that will be needed
;
DSTART: DEFS 2
CURPOS: DEFS 2
POINTR: DEFS 2
COLUMN: DEFS 2
ELMNTS: DEFS 2
CHRPTR: DEFS 1
FCB:    DEFS 32
FRATE:  DEFS 1
;
.PRINTX-INKEY-
;
; This routine uses the ROM keyboard routines to perform the
; INKEY function.  If the argument is less than 0, then the
; whole keyboard is addressed at once. If the argument is 0,
; the keyboard is scanned one time only, and the ASCII value
; is returned. If the argument is greater than zero, the keyboard
; is scanned repeatedly until a key is pressed, and its ASCII
; value is returned.
;
; HL  contains the address of the argument upon entry.
;
INKEY:
;
        CALL CLBUFF      ;clear the row history buffer
        LD A,(HL)        ;put in A for comparison
        CP 00h           ;test for - 0 +
        JP M,FSCAN       ;jump to F(ast) SCAN
        JP Z,ZSCAN       ; Z(ero) SCAN
;
        CALL 0049H       ;scan until key pressed.
        LD L,A           ;transfer A to INKEY via
        LD H,00H         ;HL register
        RET
;
ZSCAN:                   ;scan one time only
        CALL 002BH       ;and return value in A
        LD L,A
        LD H,00H
        RET
```

```
FSCAN:                          ;scan all rows at once
        LD HL,38FFH             ;including shift key
        LD A,(HL)
        LD L,A
        LD H,00H
        RET
;        CLBUFF
;
;This routine clears the row history buffer.
;
CLBUFF:
        XOR A                   ;zero A
        LD B,7                  ;there are seven buffers to zero
        LD DE,4036H             ;starting here.
BUFF:   LD (DE),A               ;clear row history buffer entry
        INC DE                  ;put next address
        DJNZ BUFF               ;do it seven times.
        RET
```

```
.PRINTX-CLS-
;
;This is the ROM CLear Screen routine, the same as writing -1C,1F- to
;the keyboard unit number.
;
CLS:    JP 01C9H
;
.PRINTX-CBL-
;
;This is the Clear Bottom Line routine, positioning the cursor to
;overwrite the bottom line on the screen.
;
CBL:    LD HL,3F80H             ;desired cursor position
        LD (4020H),HL           ;cursor storage location
        RET
```

```
.PRINTX-$INIT-
;
;This routine overides the FORLIB $INIT routine, so that the high
;memory pointer is properly set; on entry BC contains the program
;start address. A test to determine whether the
;computer is a Model I or III(4) is included.
;
$INIT:  PUSH BC           ;start of routine
        XOR A             ;zero A and F
        LD HL,$EC
        LD (HL),A
        INC HL
        LD (HL),A
        LD HL,RETADR      ;an address of a C9.
        LD ($CLSFL),HL
;
;       Test for MOD I or MOD III
;       store the HIMEM location in MEMLOC
;
        LD HL,011DH       ;ROM addr which contains either
        LD A,76           ;an "L" for Level II or and "M"
        CP (HL)           ;for Model III.
        JR Z,HMI
;
HMIII:  LD HL,(4411H)     ;MOD III HIMEM location
        LD A,62           ;">" = blinker character
        LD B,20H          ;MOD III delay counter(0=256)
        JR SPL
;
HMI:    LD HL,(4049H)     ;MOD I HIMEM location
        LD A,94           ;"->" = blinker character
        LD B,0FH          ;MOD I delay counter
;
SPL:
;
        LD DE,CHRPTR      ;store the proper blinker character
        LD (DE),A         ;where it can be retrieved.
        LD DE,FRATE       ;get the flash rate location
        LD A,B            ;transfer rate to A
        LD (DE),A         ;store it
;
        POP BC            ;restore it before we move the stack
                          ;pointer
;
        LD SP,HL          ;reset  stack pointer
        XOR A             ;clear carry flag
                          ;HL still carries the value in HIMEM
        LD DE,1080H       ;4K for displays, 128 bytes for stack
        SBC HL,DE
        LD (DSTART),HL
```

ESP Section Failure Model Listing

```
                        ;BC holds the continuation address
        LD H,B          ;put BC into HL
        LD L,C
        JP (HL)         ;start of program
;
RETADR: RET


.PRINTX -BUSY-
;
;This routine checks for printer ready.  If it is not, the routine
;POPs its return address off the stack so that when it returns,
;it returns to the prior calling program.
;
BUSY:
        LD A,(37E8H)            ;printer addr in both MODs
        CALL 05D4H              ;ROM status routine
        RET Z
;
        POP HL                  ;remove immediate RET addr
                                ;from stack.
        RET                     ;to prior caller
```

```
.PRINTX-MENU-
;
; This routine replaces the FORTRAN MENU subroutine often used
; in menu-driven programs.  On entry, HL points to the variable
; ELMNTS, DE points to COLUMN, and BC points to POINTR, all
; integer variables.
;
        VADR EQU 3CBFH          ;start of video memory for Menu
;
;
;
MENU:
        LD (ELMNTS),HL          ;save variables in order:
        LD (COLUMN),DE          ;ELMNTS, COLUMN, POINTR
        LD (POINTR),BC
;
CYCLE1: LD HL,(POINTR)
        LD A,(HL)               ;lower byte of value
        INC HL                  ;next byte
        LD H,(HL)
        LD L,A                  ;now HL = input value in POINTR
        DEC HL                  ;subtract 1
        LD B,6                  ;64 = 2**6
;
CYCLE2: ADD HL,HL               ;double it 6 times until
        DJNZ CYCLE2             ; *64 has been done
;
        LD DE,VADR              ;video offset
        ADD HL,DE               ;position in proper row
        EX DE,HL                ;store it awhile
        LD HL,(COLUMN)
        LD A,(HL)               ;lower byte of value
        INC HL                  ;next byte
        LD H,(HL)
        LD L,A                  ;now HL = input value in COLUMN
        ADD HL,DE               ;current cursor position
        LD (CURPOS),HL          ;store it till later
;
        LD A,(CHRPTR)
        LD (HL),45              ;"-"
        INC HL
        LD (HL),A               ;"->" for Mod I or III
;
        CALL BLINKR             ;the MACRO routine
;
                                ;A contains the key pressed.
        CP 13                   ;look for <Enter>
        RET Z
        PUSH AF                 ;save A register
        LD A,(FRATE)            ;start a delay routine
        LD B,A
        LD C,00
        CALL 0060H
        LD HL,(CURPOS)          ;need to blank out the arrow
        LD (HL),128             ;with a space
        INC HL
        LD (HL),128
;
        POP AF                  ;get the char from BLINKR
        CP 10                   ;"Dn Arrow"
        JR Z,DNTST
;
        CP 91                   ;"Up Arrow"
        JR Z,UPTST
;
        JR CYCLE1               ;start process over
;
DNTST:
        LD DE,(POINTR)
        LD A,(DE)
        LD HL,(ELMNTS)          ;point HL to ELMNTS
        CP (HL)
        JP P,CYCLE1             ;no change if POINTR>=ELMNTS
        EX DE,HL                ;get POINTR
        LD A,1
        ADD A,(HL)
        LD (HL),A               ;POINTR = POINTR + 1
        JR CYCLE1
;
UPTST:
        LD DE,(POINTR)
        LD A,(DE)
        CP 2
        JP M,CYCLE1             ;no change if POINTR=1
        EX DE,HL                ;get POINTR
        LD A,0FFH               ; -1
        ADD A,(HL)
        LD (HL),A               ;POINTR = POINTR - 1
        JR CYCLE1
```

ESP Section Failure Model Listing

```
.PRINTX-BLINKR-
;
; This routine replaces the FORTRAN BLINKR subroutine often used
; in menu-driven programs.  Because  it is only called from MENU,
; both MENU and BLINKR are more closely coupled, than under
; the FORTRAN calling conventions.
;
;
BLINKR:
;
        LD HL,(CURPOS)          ;current cursor position
        INC HL                  ;next byte = arrowhead
        LD (HL),128
;
NEXT1:  CALL CLBUFF             ;clear the buffer
;
        LD B,80H                ;delay count to flash slowly
;
CYCLE3: CALL 002BH              ;scan once each time for
        OR A                    ;key depression
        RET NZ                  ;go check it out for value
        DJNZ CYCLE3
;
        BIT 7,(HL)              ;test for 128 or ASCII
;
        JR Z,BLINKR             ;go to the 128 unit
;
        LD A,(CHRPTR)           ;get arrow char.
        LD (HL),A
        JR NEXT1                ;go to the next test
```

```
.PRINTX-FCHECK-
;
; FCHECK can be called from a FORTRAN program to test for the
; existence of a data file.  The syntax is:
;        CALL FCHECK(EXIST,FILN)
; where EXIST is a LOGICAL variable which is returned by
; FCHECK and FILN is a LOGICAL array which contains the name
; of the file whose existence is uncertain.  If the file is
; found then FCHECK returns EXIST=.TRUE., otherwise
; EXIST = .FALSE.
;
OPEN    EQU     4424H
FSPEC   EQU     441CH
CLOSE   EQU     4428H
;
FCHECK: PUSH    BC                      ;save all registers
        PUSH    DE
        PUSH    AF
        PUSH    HL
;
        EX      DE,HL                   ;move string pointer to HL
        LD      DE,FCB                  ;point to beginning of FCB
        CALL    FSPEC                   ;move filespec to FCB
        JR      NZ,ERROUT               ;check for error in filespec
        LD      HL,0                    ;A NONSENSE BUFFER
        CALL    OPEN                    ;OPEN EXISTING FILE
        PUSH    AF                      ;SAVE FLAG
        CALL    CLOSE                   ;CLOSE FILE
        POP     AF                      ;RECALL FLAG
;
ERROUT: POP     HL                      ;RECALL PARAMETER ADDRESS
        JR      Z,OK                    ;SKIP IF NO ERROR
        LD      A,0                     ;FORTRAN -FALSE-
        LD      (HL),A                  ;STORE IT IN PARAMETER
        JR      RETURN                  ;GOTO TO RETURN
;
OK:     LD      A,0FFH                  ;FORTRAN -TRUE-
        LD      (HL),A                  ;STORE IT IN PARAMETER
;
RETURN: POP     AF
        POP     DE
        POP     BC
        RET
;

.PRINTX -DOSCLL-
;
;This is the DOS-CALL routine from FORTRAN.  On entry, HL
;points to a buffer, terminated with a 0D.
;
DOSCLL: JP 4419H
;
;
        END
```

CHAPTER 3

ESP DYNAMIC DISPLAY MODEL

Program Author: Dr. P. A. Lawless

REVISION 1.2

ABSTRACT

This program calculates the bare penetration of an ESP according to a simplified collection process (the ESP sectional model), which takes into account sneakage and collection of dust on the plates. The user is then allowed to input a fraction of the dust that is reentrained (in one or more electrical sections) and a schedule for rapping the plates. The program then recalculates the ESP penetration for the moment of time following the rap and provides a graphical and printed presentation of the emissions and opacity as a function of time.

USER BENEFITS

- The ESP sectional model allows a quick look at the factors influencing ESP performance: electric field, particle size, SCA, and losses.
- The dynamic display model allows a quick look at the relative significance of rapping on outlet emissions and opacity.
- The effects of rapping schedule and rapping strategies can be evaluated quickly and concisely.
- The effects of long-term averages, such as several hours for mass sampling or the 6-running average for opacity, can be examined.

HARDWARE/SOFTWARE REQUIREMENTS

This program runs on a TRS-80 Model I, III, or 4 microcomputer with the NEWDOS/80 Version 2 operating system. It requires 48K of RAM and at least one disk drive and will accommodate several printers. A printer is useful for program output, but it is possible to use the program without a printer, since all releveant data are displayed on the screen. If a printer is used, it should be capable of printing 132 columns for the output.

The program is written mainly in Microsoft FORTRAN, but does reference some subroutines written in ASSEMBLER. If any routines are to be modified, the Microsoft programs F80, M80, and L80 would be required.

Since the program uses machine- and ROM- dependent coding, this version will only be expected to run on the above-mentioned TRS-80 machines, with the NEWDOS operating system.

METHOD

The penetration of an ESP is calculated section by section, with the total penetration given by the product of section penetrations:

$$P_t = P_1 \cdot P_2 \cdot P_3 \cdot \cdots \quad . \tag{3-1}$$

Each section penetration is assumed to be the sum of two terms, the uncollected fraction and the loss fraction:

$$P_n = LF + (1 - LF) \cdot P_c \quad . \tag{3-2}$$

The uncollected fraction is given by the Deutsch-Anderson (D-A) equation:

$$P_c = \exp(-w \cdot SCA) \ . \tag{3-3}$$

The steady loss fraction is given in terms of the sneakage (SN) by

$$LF = SN \ , \tag{3-4}$$

but mass reentrained is represented by the fraction RR of the mass collected on the plates since the last time the particular plate was rapped. The default values in the program (SN=RR=0.1) are those that give good results for most ESPs. Since the dynamic model represents the sectional model applied on a timed basis, the results of the sectional model are also available in the program, with LF given by

$$LF = SN + RR - SN \cdot RR \quad . \tag{3-5}$$

The SCA is the specific collection area for the electrical section being studied, not for the whole ESP.

The D-A migration velocity, w, is calculated for one particle size under the following assumptions:

- The minimum particle size encountered is 2 μm diameter.

- Field charging is assumed to account for the particle charge.
- The particle is assumed to reach the saturation charge characteristic of the electric field for the section it is in.
- The migration velocity is equal to the electric force divided by the viscous drag coefficient for that particle size.
- No Cunningham correction factor is used.

Under these assumptions, the migration velocity is given by

$$w = d \cdot \varepsilon \cdot E^2/\eta \qquad (3\text{-}6)$$

where

d = mean particle diameter
ε = free space permittivity
E = Electric field
η = the gas viscosity.

The particle diameter is assumed to change throughout the length of the ESP, although it is recalculated only at the outlet of each section. The diameter assumed at the inlet of the ESP is the mass median diameter (MMD) of the particle distribution. This is done for three reasons: it is often the only value known; it is usually large enough for all the assumptions to apply; and, under most ESP theories, all particles larger than the MMD will be collected at least as well as those at the MMD.

The particle diameter is always calculated as a penetration-weighted average of several particle diameters. For the uncollected fraction in the section, the diameter of particles emerging from the section is a weighted average between the inlet MMD and 2 μm at the outlet:

$$MMD_{out} = MMD_{inlet} \cdot p_c + 2.0 \cdot (1 - p_c) \ . \qquad (3\text{-}7)$$

Thus, if a section collects poorly, the MMD at the outlet is close to the MMD at the inlet; but if it collects well, the outlet MMD is about 2 μm. (The choice of 2 μm is based on studies of detailed models of ESP performance.)

For the fraction of particles that sneak through the section, the MMD at the outlet is the same as the MMD at the inlet. For the fraction of particles at the outlet due to rapping reentrainment, the MMD is fixed at 6 μm. These three characteristic sizes are combined at the outlet to calculate a single particle diameter for input to the next section:

$$\mathrm{MMD}_{\mathrm{total}} = (p_c \cdot \mathrm{MMD}_{\mathrm{out}} + \mathrm{SN} \cdot \mathrm{MMD}_{\mathrm{inlet}} + \mathrm{RR} \cdot 6.0)/(p_c + \mathrm{SN} + \mathrm{RR}) \quad . \quad (3\text{-}8)$$

Between rapping periods, the same expression for the weighted average is used, but with RR set to 0. It is this calculation of changing MMD that most affects the opacity calculation during the rap and sometimes causes unexpected (perhaps unrealistic) opacity effects. The approximation for the opacity calculation is that the particle diameter appropriate for opacity is the outlet MMD, but with a scattering cross section fixed for a smaller particle size. The opacity calculation is intended to indicate trends rather than accurate results.

One of the features of the program is the ability to specify the plate rapping scheme and to monitor its effects on the emission level. The program offers two mutually exclusive rapping schemes: a timing-initiated rapping or a dust-thickness-initiated rapping. Virtually all ESPs are cleaned on a timed rapping schedule, which will be the focus of this discussion. The dust layer thickness scheme is included in the program for research purposes.

The strategy of ESP rapping is to find the schedule that minimizes the opacity "spikes" that follow plate rapping. Since the first electrical section of an ESP is subjected to the highest dust load, the plates in that section become loaded much more quickly than the plates in succeeding sections. Therefore, the first section of an ESP is usually rapped most frequently. Similarly, since the dust load at the inlet to the second section of an ESP is greater than that entering succeeding sections, the plates in the second section become loaded more quickly than succeeding sections. Hence, this section is rapped more frequently than succeeding sections. Consequently, a typical rapping schedule for an ESP with four sections might be 6, 15, 30, 60. That is, section one is cleaned every 6 minutes, section two is cleaned every 15 minutes, and so on. For a large ESP with many independent flow channels, rapping a given section all at once would generate a large rapping plume, thereby generating an opacity "spike." To minimize the appearance of these spikes, the rapping of different lanes within a section can be distributed 1 minute apart. In the computer program, different lanes (or groups of lanes) are rapped 1 minute apart. Therefore, the rapping of sections follows a set schedule, and the rapping of lanes of a given section is also scheduled.

As an example, consider an ESP that has four electrical sections divided into two flow channels (A and B) as shown in Figure 3-1 and is cleaned according to the schedule mentioned earlier (i.e., 6-, 15-, 30- and 60-minute cycles). After 6 minutes of operation, section 1 of channel A would be rapped. One minute later section 1 of channel B would be rapped. At 12 minutes, section 1 of channel A would be rapped again and section 1 of channel B would be rapped a minute later. At 15 minutes into the cycle, section 2 of channel A would be rapped, followed a minute later by section 2 of channel B. This schedule is depicted in Figure 3-2. At 30 minutes into the flow cycle the first three sections of channel A would be rapped simultaneously, followed a minute later by the first three sections of channel B (since 30 minutes is a multiple of the rapping cycle time for each of these sections). At t=60 minutes, all sections in flow channel A are rapped simultaneously, followed by channel B at t=61 minutes. In this example, the rapping cycle is nominally 60 minutes long so that time t=60 minutes can also be considered as t=0. Therefore, the cleaning of sections 1, 2, 3, and 4 of channel B overlaps into the start of the next rap cycle. The cleaning of section 1 in channel A at time t=66 minutes corresponds to the cleaning at t=6 minutes.

Another feature that has been included in the program is the ability to let the dust load fluctuate randomly with time. The operator is allowed to enter a mean dust load and a normalized standard deviation so that the random fluctuations follow a normal distribution. These fluctuations have a short-term component to simulate random fluctuations in flow and a long-term component to simulate fluctuations in the boiler operation.

To complement the random load fluctuations, the program computes moving averages of the mass emissions and opacity. The operator is allowed to specify the averaging time for the mass emissions. The averaging time for the opacity average is fixed at 6 minutes. The program displays these results as well as the "instantaneous" values.

The preceding description is primarily intended as an overview of the model details. A more thorough description of the theoretical aspects of the model as well as the practical application is included in Appendix C.

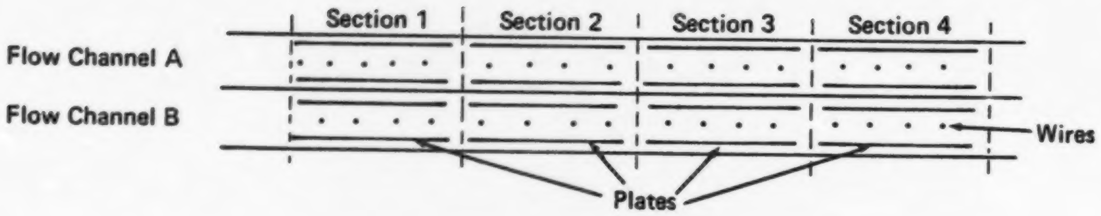
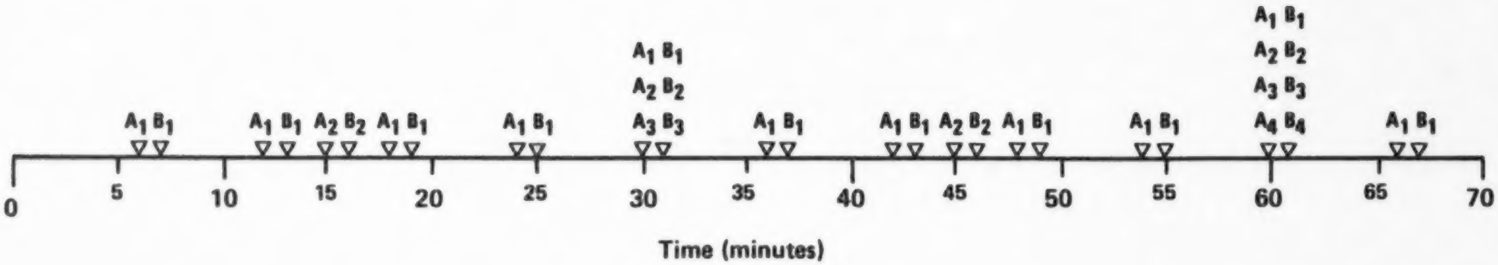


Figure 3-1. A hypothetical ESP.



A_1 Section 1 of flow channel A is rapped
A_2 Section 2 of flow channel A is rapped
A_3 Section 3 of flow channel A is rapped
A_4 Section 4 of flow channel A is rapped

B_1 Section 1 of flow channel B is rapped
B_2 Section 2 of flow channel B is rapped
B_3 Section 3 of flow channel B is rapped
B_4 Section 4 of flow channel B is rapped

Figure 3-2. A typical ESP rapping schedule.

GENERAL RESTRICTIONS

Although the program will deliver results for almost any inputs, negative values should not be used for any of the input quantities, and zero values should not be used for quantities representing physical dimensions. The results are unpredictable in those cases. The model will probably exhibit large errors if the inlet MMD is 6 μm or less, because the assumptions about particle diameter are based on experience with fly ash, and the rapping reentrainment MMD is empirically determined on fly ash ESPs. If a smaller MMD is used at the inlet, the outlet MMD will be seen to grow to a larger size than the inlet.

There is no simple way to account for back corona effects in this model. Since some empirical measurements suggest that back corona reduces particle charge by a factor of 0.5 to 0.3, the penetration through a section could be changed by a similar factor; for example, by multiplying the SCA by the same factor or by multiplying the electric field by the square root of the factor (since the penetration depends quadratically on the field.)

The best method of calculating electric field for conventional ESPs is to divide the corona electrode voltage by the electrode-plate distance and to divide the quotient by a factor of 1.5 to 1.75. Use the smaller divisor for closely spaced corona wires, or mast electrodes, and the larger divisor for more wide spaced wire electrodes.

SAMPLE PROBLEM

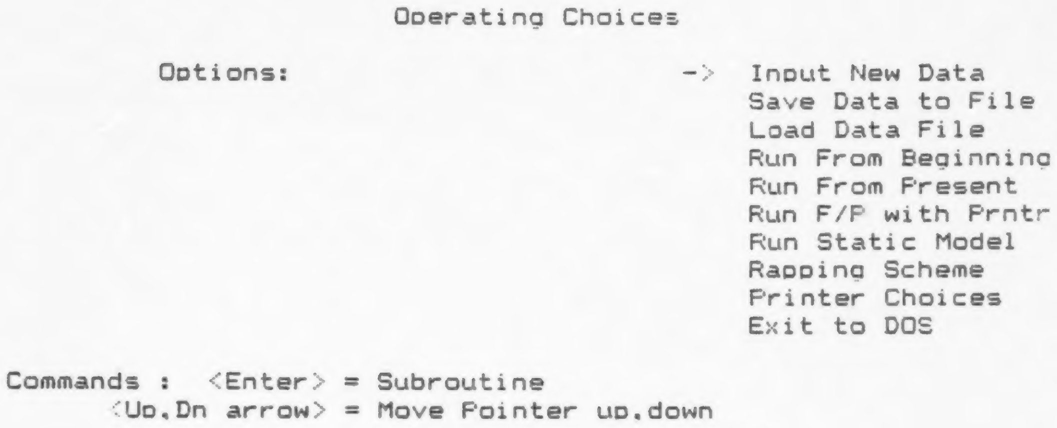
Statement: Assume you have a precipitator with an SCA of 200 ft^2/kacfm operating as a typical cold-side unit. Calculate the effects on emissions for equally spaced rapping intervals of 10 minutes.

Results: This sample problem serves two purposes. First, the user can compare his/her results with the numbers listed below to verify that the program is operating properly. Second, the step-by-step instructions given below will help the novice learn how to use the program. Since this section is intended as a tutorial, the reader is encouraged to sit in front of the computer and try each step as it is described.

When the program begins execution the title of the program will appear on the screen along with a brief description of what the program does. After these messages, the Main Menu of operating choices will appear on the screen. This menu is shown in Figure 3-3. As mentioned in Chapter 1, this menu presents a list of subroutines that can be called to perform various functions. The first step to be performed in answering the sample problem is to enter the data into the program. This can be done with the "Input New Data" option of the Main Menu. When this option is called, the menu shown in Figure 3-4 should appear on the screen. Explanations of the variables in this menu are given in Table 3-1. There are two parts to this menu. The first part consists of the first six items in the menu, which pertain to the entire ESP and/or the ESP inlet conditions. The second part of the menu begins on the line that says "For section # 1." This lower portion lists the values of sectional SCA, electric field, and rapping reentrainment factor that apply to section 1 of the ESP. Thus, if one of the first six items in this menu is changed then that new value pertains to the entire ESP. If one of the next three parameters is changed, the new value applies only to the specified section. In Figure 3-4 the parameters listed are for section 1. To view the parameters for section 2, move the blinking arrow to the menu item "Next Section" and press <ENTER>. The top portion of the menu will remain unchanged and the lower portion will show the parameters for ESP section 2. Subsequent calls to the "Next Section" command will show the parameters for the succeeding sections of the ESP. The program returns to section 1 after the last section is displayed.

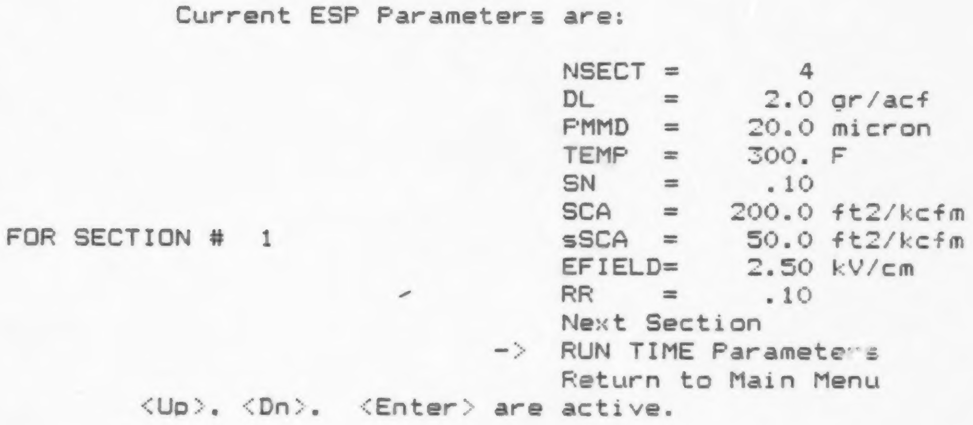
The values listed in Figure 3-4 are the default values built into the program. The SCA listed in the upper portion of the data menu is for the entire ESP. The sectional SCA listed below it (i.e., sSCA) is for the specified section. If a value for the overall SCA is entered, the program recomputes all of the sectional SCAs by dividing the overall SCA by the number of sections present. On the other hand, if a value for a sectional SCA is entered, the overall SCA will be computed as the sum of the sectional SCAs. For the sample problem, the SCA was specified as 200 ft^2/kacfm, which is the same as the default, so there is no need to change the current value.

The parameters discussed so far are all directly related to the design and operation of the ESP, hence the title "Current ESP Parameters..." at the top of each menu page. There is another set of input parameters associated with the dynamic model. These can be viewed by calling the "RUN TIME Parameters" option in the data input menu. When this function is called, the



```
                         Operating Choices

        Options:                             ->  Input New Data
                                                 Save Data to File
                                                 Load Data File
                                                 Run From Beginning
                                                 Run From Present
                                                 Run F/P with Prntr
                                                 Run Static Model
                                                 Rapping Scheme
                                                 Printer Choices
                                                 Exit to DOS

Commands :  <Enter> = Subroutine
     <Up,Dn arrow> = Move Pointer up,down
```

Figure 3-3. Main Menu of ESP dynamic model.



```
          Current ESP Parameters are:

                                    NSECT =       4
                                    DL    =     2.0 gr/acf
                                    PMMD  =    20.0 micron
                                    TEMP  =    300. F
                                    SN    =     .10
                                    SCA   =   200.0 ft2/kcfm
  FOR SECTION #  1                  sSCA  =    50.0 ft2/kcfm
                                    EFIELD=    2.50 kV/cm
                                    RR    =     .10
                                    Next Section
                                 -> RUN TIME Parameters
                                    Return to Main Menu
        <Up>, <Dn>,  <Enter> are active.
```

Figure 3-4. ESP parameters input menu.

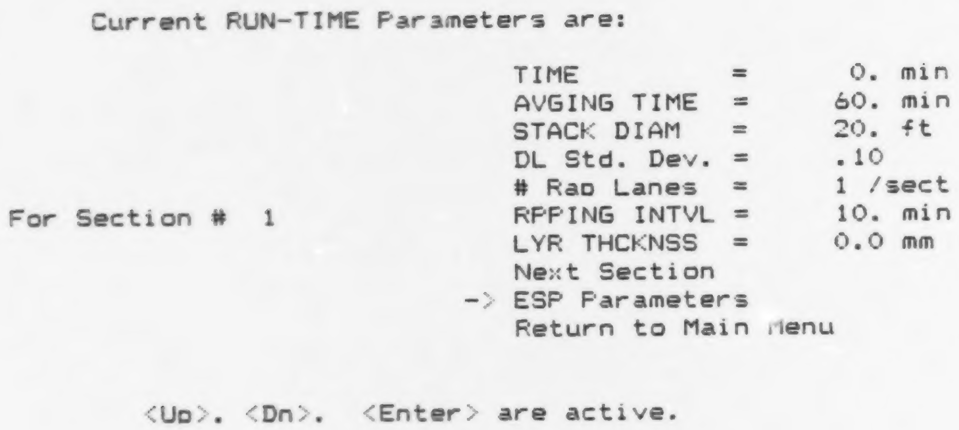
TABLE 3-1. EXPLANATION OF VARIABLES USED IN DATA MENUS

Variable	Definition	Units
ESP Parameters		
NSECT	Number of sections	
DL	Dust loading	gr/acf
PMMD	Particle mass median diameter	micrometers
TEMP	Temperature	°F
SN	Sneakage	
SCA	Specific collection area	ft^2/kacfm
sSCA	Sectional SCA	ft^2/kacfm
EFIELD	Electric field	kV/cm
RR	Rapping reentrainment factor	
RUN TIME Parameters		
TIME	Current "simulation" time	minutes
AVGING TIME	Averaging time for mass emissions	minutes
STACK DIAM	Stack diameter	feet
DL Std. Dev.	Normalized standard deviation of the dust load	
# Rap Lanes	Number of rap lanes	per section
RPPING INTVL	A section's rapping interval	minutes
LYR THCKNSS	Dust layer thickness	millimeters

menu shown in Figure 3-5 should appear on the screen. Again refer to Table 3-1 for variable explanations. This menu is similar to the menu shown in Figure 3-4; it is divided into two portions with the upper half displaying parameters that apply to the overall ESP and the lower half displaying section parameters. The TIME parameter at the top of the screen is the current "simulation" time and is included so that the operator can know how many "minutes" have elapsed in the simulation. This parameter is included for those special cases where it is necessary to change the simulation time without affecting the accumulated averages.

Below the "TIME" parameter is "AVGING TIME." This is an input parameter that specifies the period over which the running average of mass emissions is computed. In the default case, the averaging time is 60 minutes, so that the emissions average at time t is computed as the average value for the preceding 60 minutes. The "STACK DIAM" parameter is used in the calculation of opacity. The parameter "DL Std. Dev." is the normalized standard deviation of the dust load. As mentioned above, the dust load is allowed to fluctuate randomly about the mean value. These fluctuations are forced to follow a normal distribution and the "DL Std. Dev." parameter allows the user to specify how large the fluctuations are. A value of zero eliminates the fluctuations completely. The normalized standard deviation (also known as the coefficient of variation) is the ratio of the standard deviation to the mean.

The last parameter in the upper portion of the menu is "# Rap Lanes" (number of rap lanes). This is the number of subintervals that are to be used for rapping each section. For example, in Figure 3-1 there were two independent flow channels (A and B) and they were rapped 1 minute apart (see Figure 3-2). Thus there were two rapping lanes. If a third flow channel (C) were added to the ESP and it was rapped 1 minute after channel B, then the ESP would have three rap lanes. It is not necessarily true that the number of rap lanes is the same as the number of flow channels. For example, if there are six flow channels in an ESP (labeled A, B, C, D, E, and F) and two channels were rapped together, there would be only three rap lanes. In practice, this staggering of flow channel rapping is preferred because it reduces opacity "spikes." For purposes of the sample problem "# Rap Lanes" should be set to one.



```
        Current RUN-TIME Parameters are:

                                          TIME          =       0. min
                                          AVGING TIME   =      60. min
                                          STACK DIAM    =      20. ft
                                          DL Std. Dev.  =      .10
                                          # Rap Lanes   =      1 /sect
    For Section #  1                      RPPING INTVL  =      10. min
                                          LYR THCKNSS   =      0.0 mm
                                          Next Section
                                       -> ESP Parameters
                                          Return to Main Menu


           <Up>. <Dn>.  <Enter> are active.
```

Figure 3-5. RUN-TIME parameters input menu.

The section-specific parameters in the menu of Figure 3-5 are "RPPING INTVL" and "LYR THCKNSS." The "RPPING INTVL" is the rapping interval for this section. For example, in the default menu, section 1 is rapped every 6 minutes. This is one of the parameters to be changed for the sample problem. Enter a value of 10.0 minutes here. The "LYR THCKNSS" (layer thickness) shows the amount of dust that has been accumulated on the plates in the section. This parameter is included mostly for information purposes, but it can also be used to preset a layer thickness on a plate before rapping. That is, as the simulation progresses, the "LYR THCKNSS" parameter is updated to show the amount of dust collected on the plate. By changing this value the user can speed up or delay the onset of rapping for that section.

The last three options in the "RUN-TIME Parameters" menu provide access to other menus. By calling the "Next Section" option, the run-time parameters for section 2 will be displayed. As with the ESP parameter menu, successive calls to the "Next Section" command will display the parameters for each section, while leaving the other parameters unchanged. For the sample problem, call up each section menu and change the "RPPING INTVL" to 10 minutes for each. The ESP parameter menu can be recalled at any time by calling the "ESP Parameters" function of the run-time menu. To exit from either the run-time or the ESP parameter menu, use the "Return to Main Menu" function.

The "Input New Data" subroutine is quite large, with several different submenus. The block diagram in Figure 3-6 can be used as a guide for finding the different menus as they are needed.

The next step in solving the sample problem is to specify the rapping scheme. This can be accomplished with the "Rapping Scheme" option of the Main Menu. Figure 3-7 shows the menu that appears when this function is called. In the figure, "Timed Rapping" is listed as "<ACTV>" (active). This means that the program will simulate a timed rapping sequence. If "Thickness Rapping" had been chosen then the user could also enter a value for the "Critical Thickness," the thickness at which rapping would be triggered. The program is designed with timed rapping as the assumed mode of cleaning; therefore, this subroutine can usually be ignored, since most modeling will be done with timed rapping.

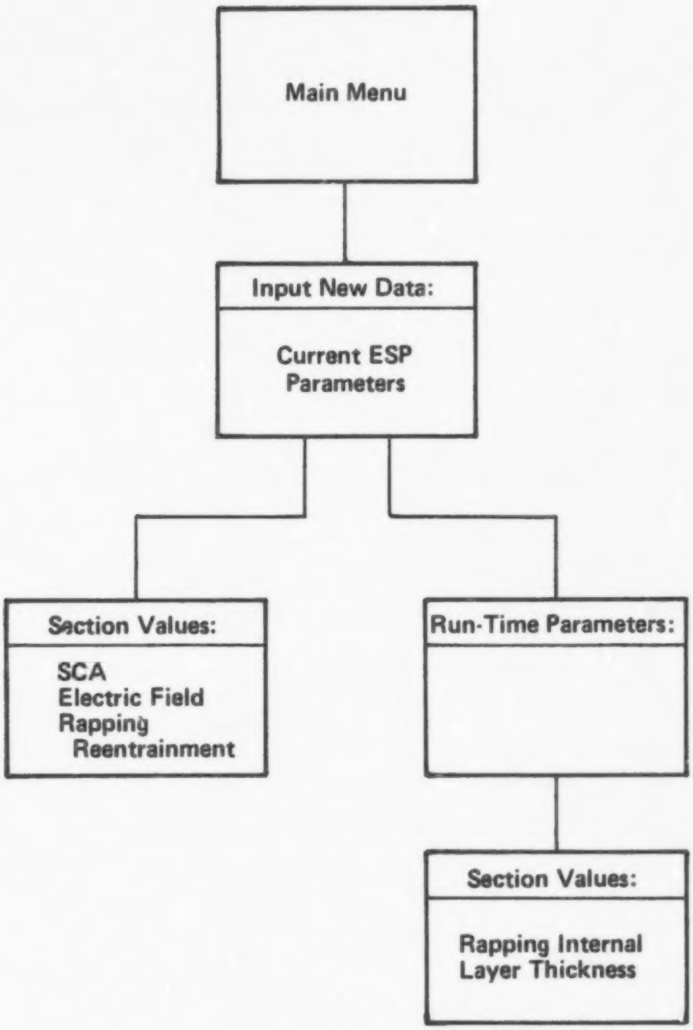
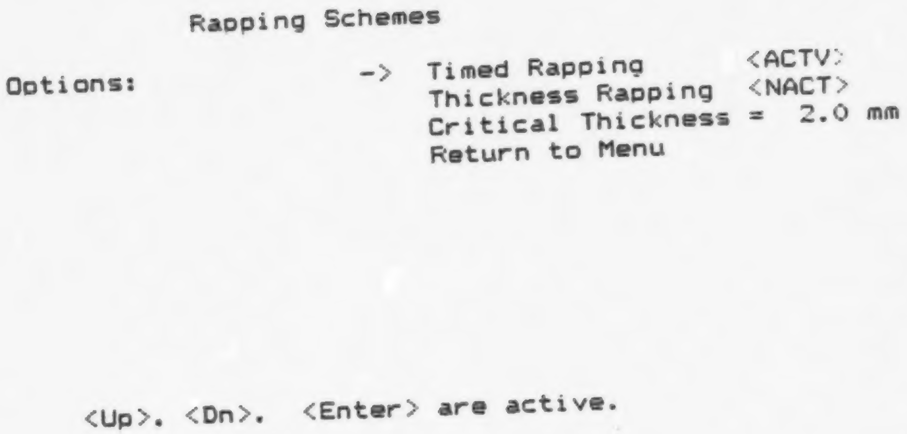


Figure 3-6. Menu organization.



```
            Rapping Schemes

Options:                ->  Timed Rapping       <ACTV>
                            Thickness Rapping   <NACT>
                            Critical Thickness =  2.0 mm
                            Return to Menu




    <Up>. <Dn>.  <Enter> are active.
```

Figure 3-7. Rapping schemes option menu.

Once the input parameters have been entered, the modeling calculations can begin. Choose the "Run From Beginning" option from the Main Menu. The screen will be cleared and a set of graphics will continually move across the screen from right to left, gradually filling it. After a few seconds, the screen will appear as in Figure 3-8. The intent of the display is to simulate a strip-chart recorder as it records data. The pen of the recorder is at the right margin of the screen. The example in Figure 3-8 is a "snapshot" of the recorder output. The "paper" of the recorder extends from the broken horizontal line near the bottom of the display to the broken horizontal line near the top of the display. Each increment of the horizontal scale corresponds to 1 minute of simulation time. The data being recorded in this display are the mass emissions from the ESP and the scale is from 0 gr/acf at the bottom of the display to 0.05 gr/acf at the top of the display. At the upper right corner of the display is the message "<Rapped 2>". This means that the most recently rapped section is number 2. In reality, this display corresponds to a two-pen chart recorder. That is, there are two sets of data being displayed: the instantaneous mass emissions and the running average mass emissions. The instantaneous mass emissions follow a pattern similar to that shown in Figure 3-9. The averaged emissions follow a pattern similar to that shown in Figure 3-10. When these images are overlaid (and discretized) then the image in Figure 3-8 results.

At the bottom of the display is a list of single letter commands that can be executed at any time. For example, to return to the Main Menu, press the <M> key on the keyboard. To make the display move faster, hold down the <F> key for a few seconds. To slow down the display hold down the <S> key for a few seconds. To proceed to the next display, hold down the <N> key, and, to begin recording the "strip chart" on a printer, hold down the <P> key. The program will send the results to the printer as they are displayed as long as the <P> key is held down. To stop printing release the <P> key.

As mentioned above, the <N> command will shift the program to the next display. There are a total of four different displays. The second display shows another strip chart record of the average mass emissions but with a higher resolution (0-0.02 gr/acf). In display number 2, only the average emissions are shown, so this can be considered as a single-pen recorder. Pressing the <N> key another time will bring the third display to the screen.

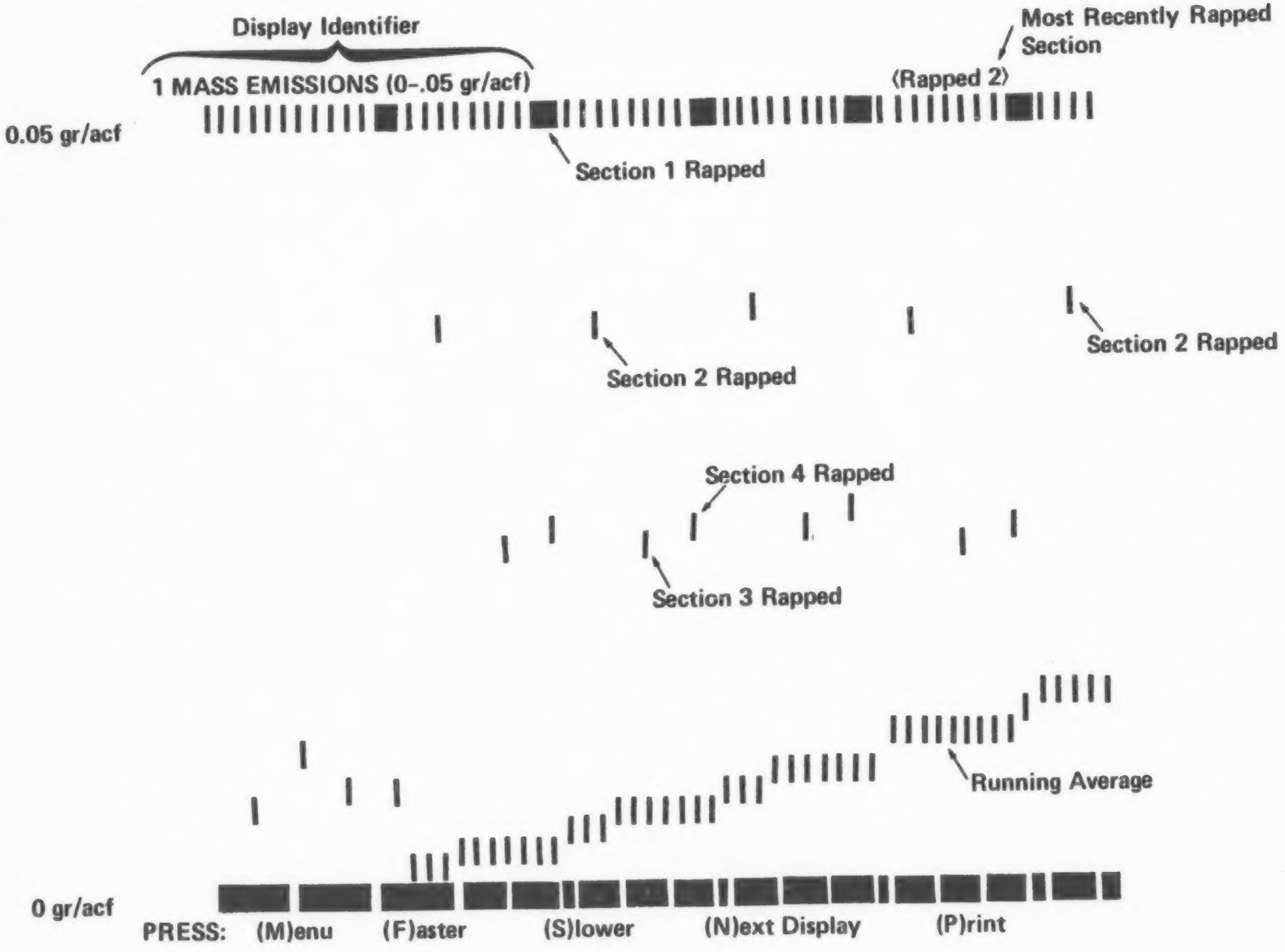


Figure 3-8. Strip-Chart display.

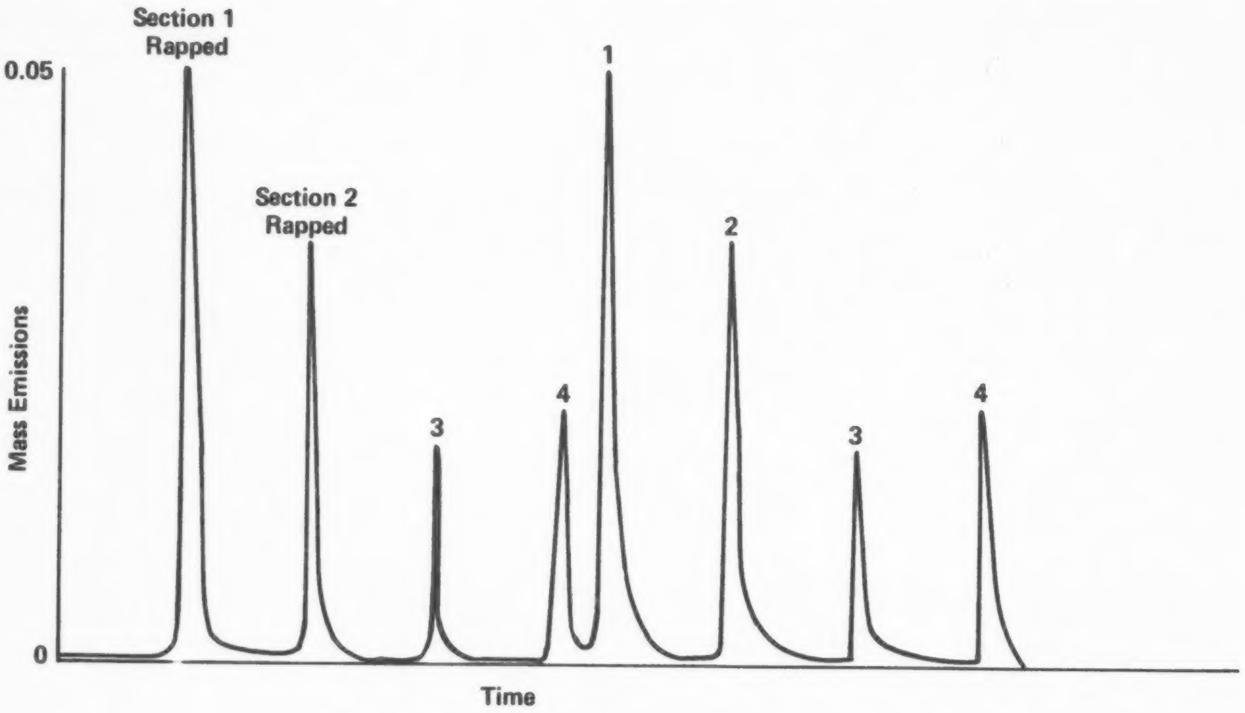


Figure 3-9. "Instantaneous" values of mass emissions.

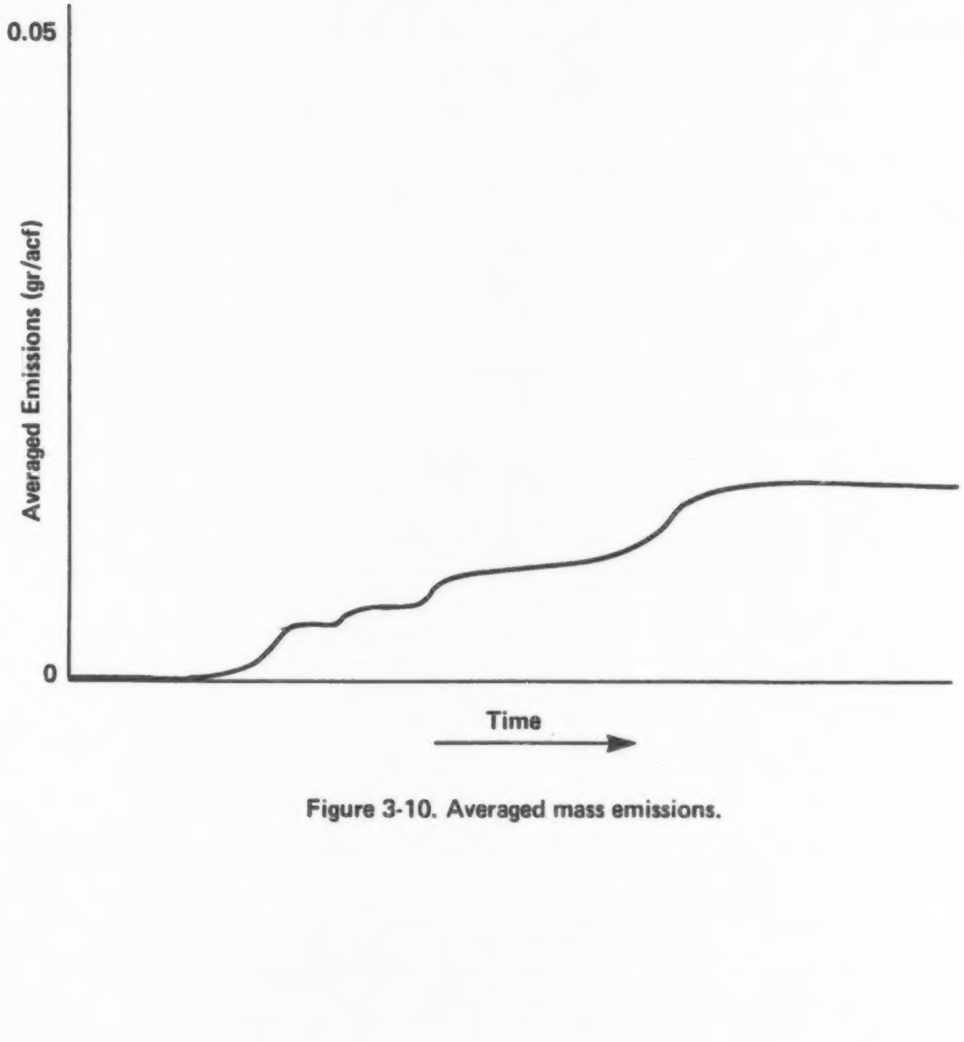


Figure 3-10. Averaged mass emissions.

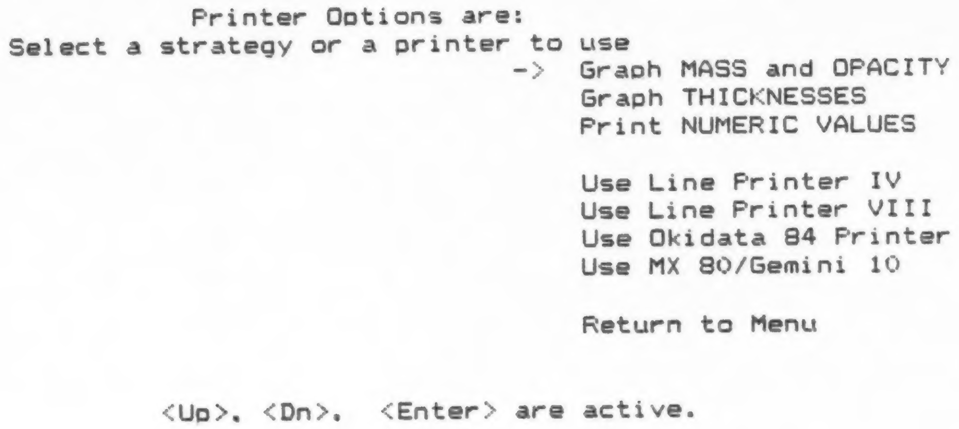
This display shows the instantaneous opacity trace with a scale of 0 to 95 percent opacity. The last display (number 4) shows the 6-minute running average opacity on a scale of 0 to 50 percent. If the <N> key is pressed while display number 4 is being shown, the program goes back to display number 1 (mass emissions).

The results shown in Figure 3-8 are characteristic of the start-up of the program. After several averaging-time cycles have elapsed (as specified in the RUN-TIME parameters input menu) the display becomes characteristic of steady-state operation. At this point, the random fluctuations that occur in mass emissions and opacity are a direct result of the fluctuations in the dust load.

Returning to the sample problem, the objective here is to see how the chosen rapping cycle affects the emissions. The graphic display provides a quick way to see the results but a hardcopy printout is needed to keep a record. For this purpose, the program is designed to work in conjunction with a printer. Before the printer output is begun it is necessary to tell the program what type of printer is being used with the "Printer Choices" option of the Main Menu. This menu, shown in Figure 3-11, consists of two parts: the output selection and the printer selection. The output selection allows the user to choose the form and content of the output. That is, the "Graph MASS and OPACITY" option specifies that the output will be in the form of a graph (or strip chart) of the instantaneous and time-averaged values of mass emissions and opacity. The "Graph THICKNESSES" option specifies graphical output of the dust thickness in each section as a function of time. The "Print NUMERIC VALUES" option specifies tabulated values of the instantaneous and time-averaged values of mass emissions and opacity as well as the instantaneous values of dust layer thickness in each section. Note that there will be no noticeable effect when these options are chosen; the choice will become evident later. For the sample problem, choose the "Graph MASS and OPACITY" option.

The second selection in the "Print Options" menu is the printer type. Choose one of the printers included in the list and then return to the Main Menu.

The program is now set to provide both screen and hardcopy output. From the Main Menu call the "Run F/P with Prntr" option (Run From Present



```
                    Printer Options are:
     Select a strategy or a printer to use
                                   ->  Graph MASS and OPACITY
                                       Graph THICKNESSES
                                       Print NUMERIC VALUES

                                       Use Line Printer IV
                                       Use Line Printer VIII
                                       Use Okidata 84 Printer
                                       Use MX 80/Gemini 10

                                       Return to Menu


           <Up>, <Dn>,  <Enter> are active.
```

Figure 3-11. Printer options menu.

with Printer). The program will print a heading on the printer and then begin displaying the "strip-chart" on the screen. For every time step displayed on the screen, the program will graph that same time step on the printer. An example printout is shown in Figure 3-12. The program is designed to retain the current simulation time and all of the modeling parameters, even though the user may return to the Main Menu. Therefore, the command to "Run From Present" means that the program will restart the calculations from the point where it left off. In Figure 3-12, the program restarts the calculations at time t=1210 minutes because that is where it was stopped. Figures 3-13 and 3-14 show sample printouts that result when the "Graph THICKNESSES" and "Print NUMERIC VALUES" options of the "Printer Options" menu are specified. These printouts provide a record of the program results and can be used to compare the effects of changing various input parameters.

The "Run Static Model" option of the Main Menu has been included for purposes of comparison. The static model is the same one used in the Section Failure model described in Chapter 2. This model corresponds to a "continuously cleaned" ESP and represents a limiting case of ESP performance. The more frequent the cleaning in the dynamic model (as specified by the "RPPING INTVL" or the "Critical Thickness" parameters), the closer will be the results of the dynamic and static models. Figure 3-15 shows the hardcopy printout from the static model based on the sample problem input values.

The remaining two options in the Main Menu are the "Save Data to File" and "Load Data File" functions. These options are useful when one particular ESP configuration is to be used several times. Rather than typing in the input data every time the program is run, the values can be entered once and saved in a data file on a floppy diskette using the "Save ..." function. Then, at some point in the future when that same ESP configuration is needed, the "Load..." function can be used to recall the data from the disk file. Figure 3-16 shows the message that appears on the screen when the "Save..." function is called. A similar message appears for the "Load..." function. Refer to your DOS manual for rules on creating valid file names.

In summary, the program gives the user the ability to model ESP operation with a wide range of cleaning schedules. As such, the program is a tool to help determine what cleaning schedule produces the lowest emissions

```
                          DATA FOR THE REAL TIME RUNNING MODEL
                               Precipitator Parameters:

Num Sect =  4  Total SCA =  200. ft2/kACFM  Sneakage = .10 /sect  Rap Lanes =  1 /sec  Temp =  300. Deg F.
              Sectional SCA's =    50.0   50.0   50.0   50.0
         Sect Fields (kV/cm)  =    2.50   2.50   2.50   2.50
                  Rap Factors =     .10    .10    .10    .10
               Rapper Periods =    10.    10.    10.    10.
             Layer Thickness  =     .89    .06    .00    .01
 Dust Load =  2.0 gr/ACF  DL S.D.= .10  Part MMD = 20.0  Stack Diam = 20. ft  Time = 1210. min  Avging Per =  60. min
```

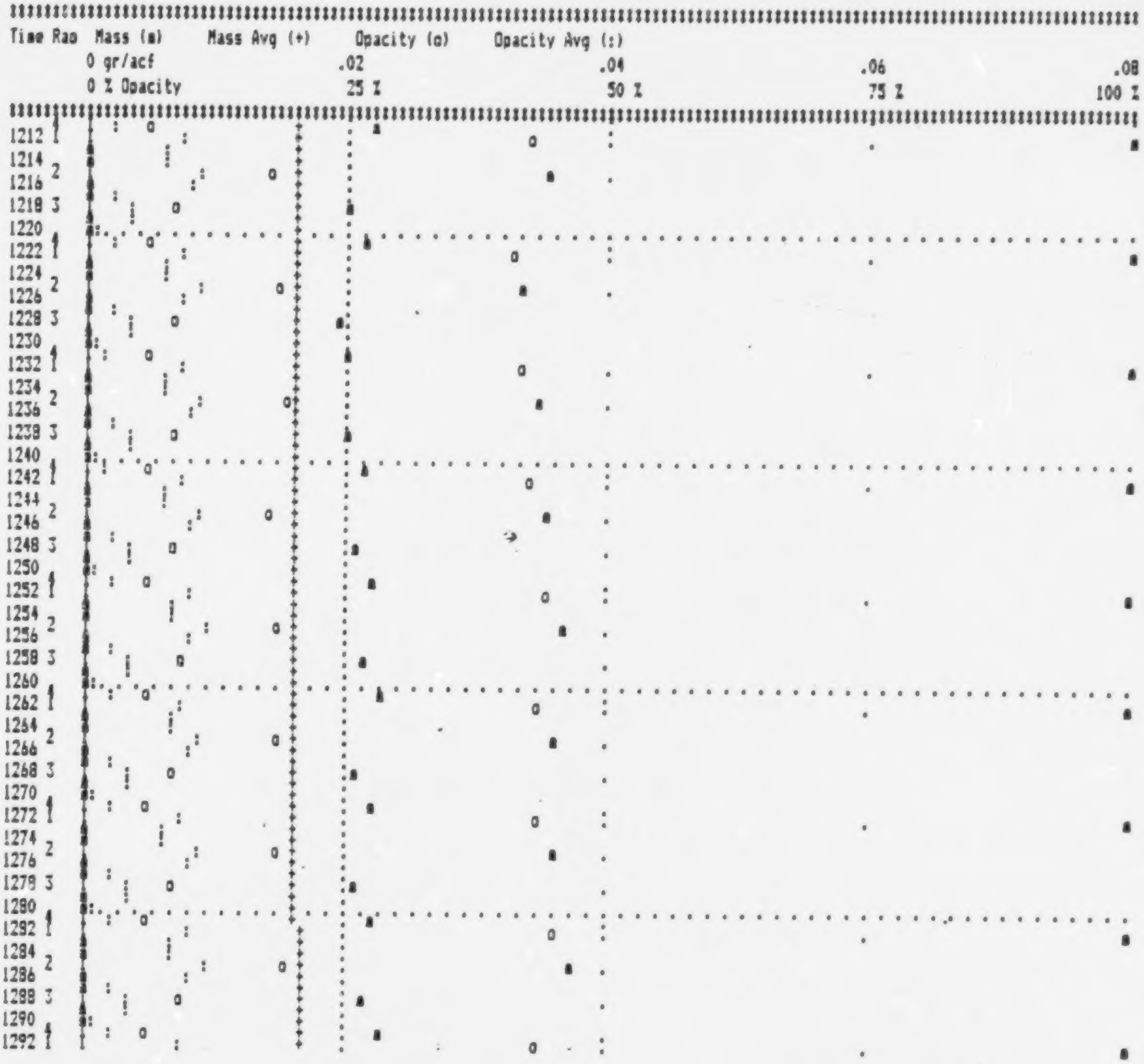


Figure 3-12. Strip-Chart printout.

```
                    DATA FOR THE REAL TIME RUNNING MODEL
                          Precipitator Parameters:

Num Sect =  4  Total SCA =  200. ft2/kACFM  Sneakage = .10 /sect  Rap Lanes =  1 /sec  Temp =  300. Deg F.
              Sectional SCA's =    50.0   50.0   50.0   50.0
         Sect Fields (kV/cm)  =    2.50   2.50   2.50   2.50
                 Rap Factors =      .10    .10    .10    .10
              Rapper Periods =     10.    10.    10.    10.
             Layer Thickness  =     .73    .04    .00    .01
 Dust Load =  2.0 gr/ACF  DL S.D.= .10  Part MMD = 20.0  Stack Diam = 20. ft  Time = 1339. min  Avging Per =  60. min
```

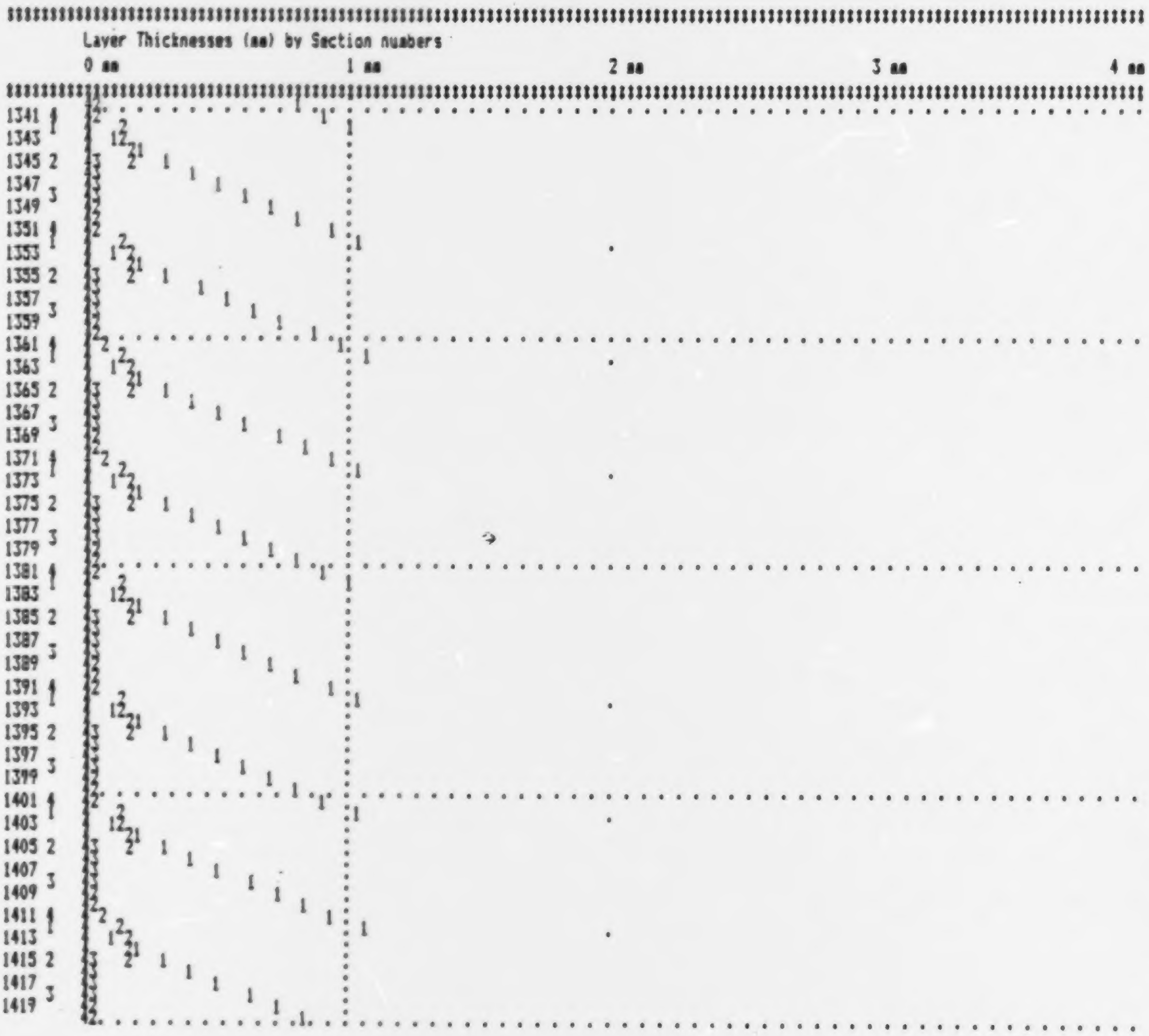
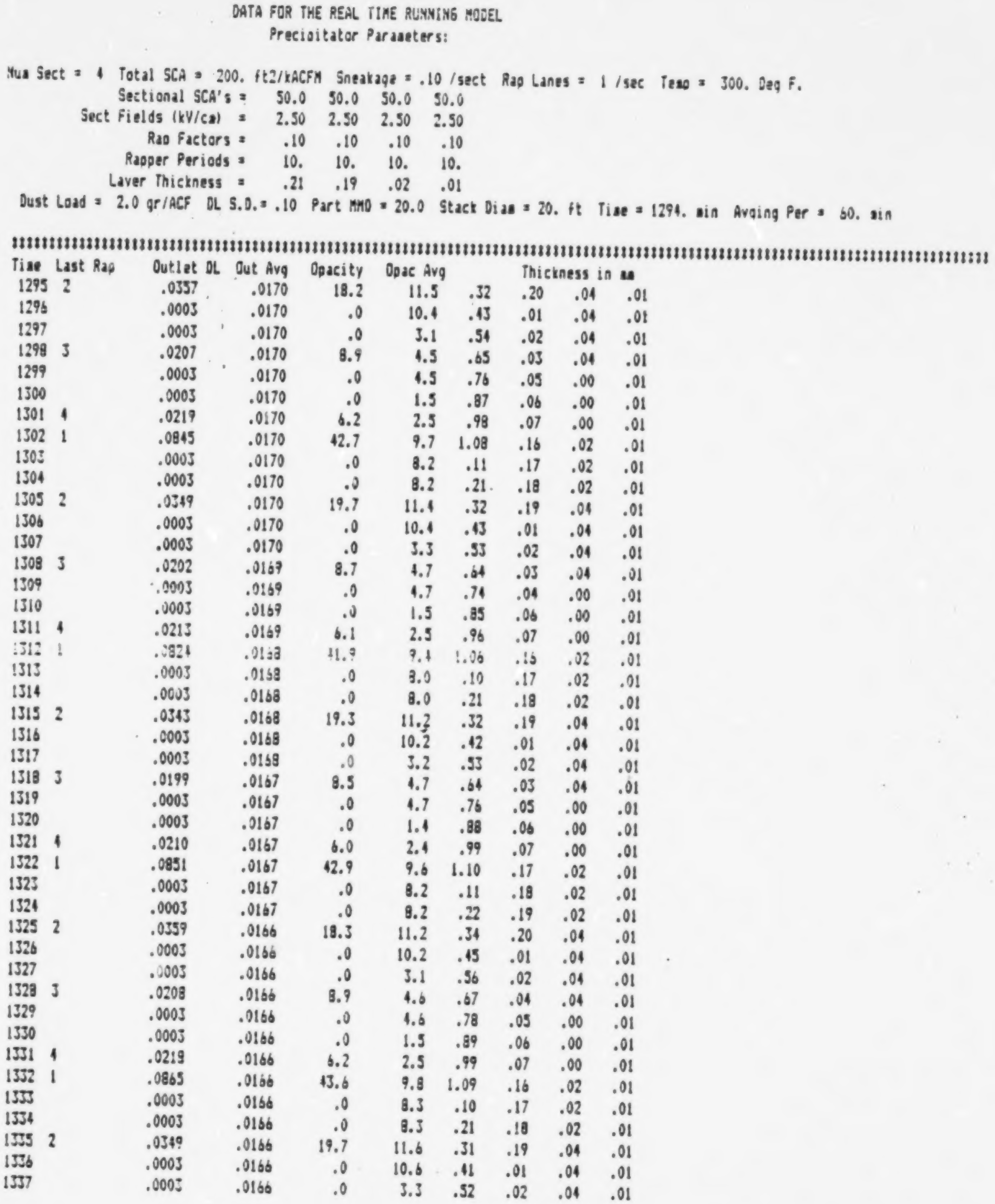
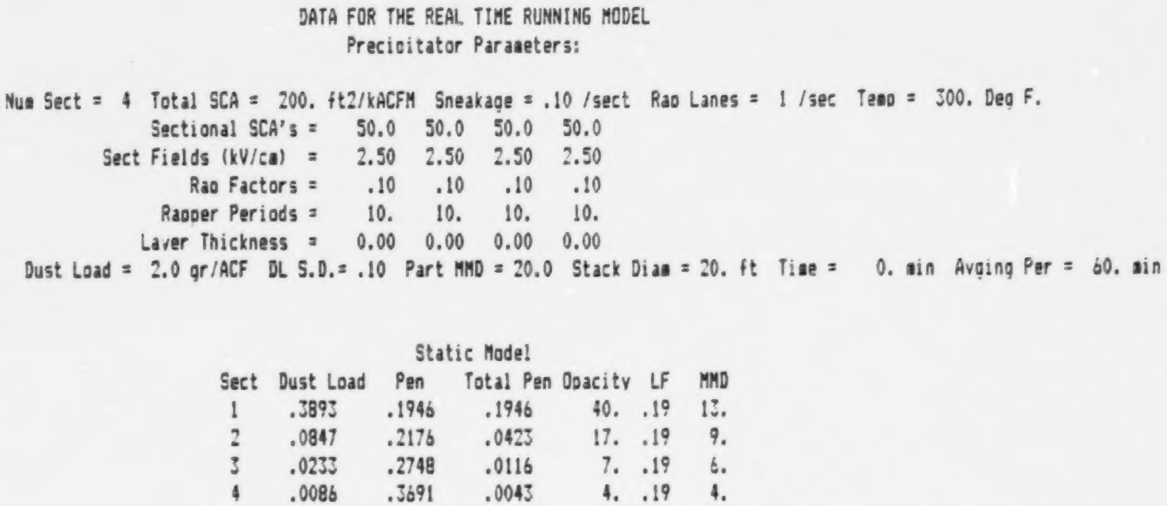


Figure 3-13. "Graph Thickness" printout.



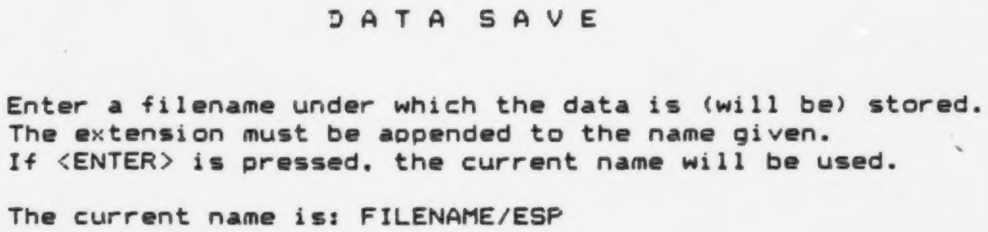
```
                         DATA FOR THE REAL TIME RUNNING MODEL
                              Precipitator Parameters:

Num Sect =  4  Total SCA =  200. ft2/kACFM  Sneakage = .10 /sect  Rap Lanes =  1 /sec  Temp =  300. Deg F.
               Sectional SCA's =    50.0   50.0   50.0   50.0
          Sect Fields (kV/cm)  =    2.50   2.50   2.50   2.50
                   Rap Factors =     .10    .10    .10    .10
                Rapper Periods =     10.    10.    10.    10.
               Layer Thickness =     .21    .19    .02    .01
 Dust Load =  2.0 gr/ACF  DL S.D.= .10  Part MMD = 20.0  Stack Diam = 20. ft  Time = 1294. min  Avging Per =  60. min

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
Time  Last Rap    Outlet DL  Out Avg   Opacity   Opac Avg           Thickness in mm
 1295  2            .0357     .0170      18.2       11.5    .32    .20    .04    .01
 1296               .0003     .0170        .0       10.4    .43    .01    .04    .01
 1297               .0003     .0170        .0        3.1    .54    .02    .04    .01
 1298  3            .0207     .0170       8.9        4.5    .65    .03    .04    .01
 1299               .0003     .0170        .0        4.5    .76    .05    .00    .01
 1300               .0003     .0170        .0        1.5    .87    .06    .00    .01
 1301  4            .0219     .0170       6.2        2.5    .98    .07    .00    .01
 1302  1            .0845     .0170      42.7        9.7   1.08    .16    .02    .01
 1303               .0003     .0170        .0        8.2    .11    .17    .02    .01
 1304               .0003     .0170        .0        8.2    .21    .18    .02    .01
 1305  2            .0349     .0170      19.7       11.4    .32    .19    .04    .01
 1306               .0003     .0170        .0       10.4    .43    .01    .04    .01
 1307               .0003     .0170        .0        3.3    .53    .02    .04    .01
 1308  3            .0202     .0169       8.7        4.7    .64    .03    .04    .01
 1309               .0003     .0169        .0        4.7    .74    .04    .00    .01
 1310               .0003     .0169        .0        1.5    .85    .06    .00    .01
 1311  4            .0213     .0169       6.1        2.5    .96    .07    .00    .01
 1312  1            .0824     .0168      41.9        9.4   1.06    .16    .02    .01
 1313               .0003     .0168        .0        8.0    .10    .17    .02    .01
 1314               .0003     .0168        .0        8.0    .21    .18    .02    .01
 1315  2            .0343     .0168      19.3       11.2    .32    .19    .04    .01
 1316               .0003     .0168        .0       10.2    .42    .01    .04    .01
 1317               .0003     .0168        .0        3.2    .53    .02    .04    .01
 1318  3            .0199     .0167       8.5        4.7    .64    .03    .04    .01
 1319               .0003     .0167        .0        4.7    .76    .05    .00    .01
 1320               .0003     .0167        .0        1.4    .88    .06    .00    .01
 1321  4            .0210     .0167       6.0        2.4    .99    .07    .00    .01
 1322  1            .0851     .0167      42.9        9.6   1.10    .17    .02    .01
 1323               .0003     .0167        .0        8.2    .11    .18    .02    .01
 1324               .0003     .0167        .0        8.2    .22    .19    .02    .01
 1325  2            .0359     .0166      18.3       11.2    .34    .20    .04    .01
 1326               .0003     .0166        .0       10.2    .45    .01    .04    .01
 1327               .0003     .0166        .0        3.1    .56    .02    .04    .01
 1328  3            .0208     .0166       8.9        4.6    .67    .04    .04    .01
 1329               .0003     .0166        .0        4.6    .78    .05    .00    .01
 1330               .0003     .0166        .0        1.5    .89    .06    .00    .01
 1331  4            .0219     .0166       6.2        2.5    .99    .07    .00    .01
 1332  1            .0865     .0166      43.6        9.8   1.09    .16    .02    .01
 1333               .0003     .0166        .0        8.3    .10    .17    .02    .01
 1334               .0003     .0166        .0        8.3    .21    .18    .02    .01
 1335  2            .0349     .0166      19.7       11.6    .31    .19    .04    .01
 1336               .0003     .0166        .0       10.6    .41    .01    .04    .01
 1337               .0003     .0166        .0        3.3    .52    .02    .04    .01
```

Figure 3-14. "Print NUMERIC VALUES" printout.



```
                         DATA FOR THE REAL TIME RUNNING MODEL
                              Precipitator Parameters:

Num Sect =  4  Total SCA =  200. ft2/kACFM  Sneakage = .10 /sect  Rap Lanes =  1 /sec  Temp =  300. Deg F.
               Sectional SCA's =    50.0   50.0   50.0   50.0
          Sect Fields (kV/cm)  =    2.50   2.50   2.50   2.50
                   Rap Factors =     .10    .10    .10    .10
                Rapper Periods =     10.    10.    10.    10.
               Layer Thickness  =   0.00   0.00   0.00   0.00
 Dust Load =  2.0 gr/ACF  DL S.D.= .10  Part MMD = 20.0  Stack Diam = 20. ft  Time =    0. min  Avging Per =  60. min


                                   Static Model
                      Sect  Dust Load   Pen    Total Pen Opacity  LF   MMD
                       1      .3893    .1946    .1946      40.  .19   13.
                       2      .0847    .2176    .0423      17.  .19    9.
                       3      .0233    .2748    .0116       7.  .19    6.
                       4      .0086    .3691    .0043       4.  .19    4.
```

Figure 3-15. Static model printout.



```
                    D A T A   S A V E


Enter a filename under which the data is (will be) stored.
The extension must be appended to the name given.
If <ENTER> is pressed, the current name will be used.

The current name is: FILENAME/ESP
```

Figure 3-16. "Save Data" file name request.

and opacity. The user can specify the important design parameters for the ESP and view a simulation of the resulting emissions and opacity while the dust load fluctuates randomly. The results can also be directed to the printer in either graphical or numeric form The Static Model can be used to compute the results for an ESP of similar design with continuous rapping. To make the program more convenient to use, the program allows the user to save and load the ESP design parameters in data files and to specify the type of printer that is being used with the microcomputer.

```
      PROGRAM ESPDRV

C THIS PROGRAM DISPLAYS THE MENUS FOR ALL THE DATA OPTIONS.
C THE BLOCK DATA PROGRAM "ESPDAT" CONTAINS THE DEFAULT VALUES.

       INTEGER POINTR,COLUMN,ELMNTS
       EXTERNAL ESPDAT

C PUT UP A SCREEN MESSAGE AND WAIT FOR REPLY

       CALL SCRNMS

       CALL CLS

C ALLOW ABOUT 100 BYTES FOR THE STACK BELOW HIMEM, THEN
C PROTECT 4K OF HIGH MEMORY, FILL IT WITH BLANKS, AND
C WRITE TITLES IN EACH 1K SEGMENT.

       CALL FILL

       CALL TITLES

       POINTR = 1

1100   CONTINUE

C CLEAR SCREEN

       CALL CLS

       ELMNTS = 10
       COLUMN = 41

       WRITE(3,102)
102    FORMAT('+',23X,'Operating Choices'
     * //'+',2X,'         Options:       ',22X,'Input New Data'
     * /'+',44X,'Save Data to File'
     * /'+',44X,'Load Data File'
     * /'+',44X,'Run From Beginning'
     * /'+',44X,'Run From Present'
     * /'+',44X,'Run F/P with Prntr'
     * /'+',44X,'Run Static Model'
     * /'+',44X,'Rapping Scheme'
     * /'+',44X,'Printer Choices'
     * /'+',44X,'Exit to DOS'
     * //'+Commands :   <Enter> = Subroutine'
     * /'+         <Up,Dn arrow> = Move Pointer up,down')

       CALL MENU(ELMNTS,COLUMN,POINTR)

       GOTO (1300,1500,1700.1900,2100,2300,2500,
     * 2700,2900,3100),POINTR
```

```
1300    CALL DATAIN
        GO TO 1100

1500    CALL SAVE
        GO TO 1100

1700    CALL LOAD
        GO TO 1100

1900    CALL RUNEP0
        GO TO 1100

2100    CALL RUNEP
        GO TO 1100

2300    CALL PRUNEP
        GO TO 1100

2500    CALL RUNSTT
        GO TO 1100

2700    CALL RAPIN
        GO TO 1100

2900    CALL PRTOPT
        GO TO 1100

3100    CALL CLS
        STOP

        END
```

```
      SUBROUTINE DATAIN

       INTEGER COLUMN,POINTR,ELMNTS
       COMMON /EPPAR/NSECT,DL,PMMD,TEMP,SN,SCA
       COMMON /SECTL/SSCA(10),EFLD(10),RR(10)
       COMMON /RTME/TIME,TAVG,RAPT(10),THD(10),DLDEV,STCKDM,NLNS

 700   ISECT = 1

C CLEAR THE SCREEN.

       POINTR = 11

 300   CALL CLS

       WRITE(1,3) NSECT,DL,PMMD,TEMP,SN,SCA,ISECT,
     * SSCA(ISECT),EFLD(ISECT),RR(ISECT)
   3   FORMAT('+',15X,'Current ESP Parameters are:'//
     * '+',38X,'NSECT =',I8/
     * '+',38X,'DL    =',F8.1,' gr/acf'/
     * '+',38X,'PMMD  =',F8.1,' micron'/
     * '+',38X,'TEMP  =',F8.0,' F'/
     * '+',38X,'SN    =',F8.2/
     * '+',38X,'SCA   =',F8.1,' ft2/kcfm'/
     * '+',5X,'FOR SECTION # ',I2,17X,'sSCA  =',F8.1,
     *    ' ft2/kcfm'/
     * '+',38X,'EFIELD=',F8.2,' kV/cm'/
     * '+',38X,'RR    =',F8.2/
     * '+',38X,'Next Section'/
     * '+',38X,'RUN TIME Parameters'/
     * '+',38X,'Return to Main Menu')

       CALL CBL

       WRITE(1,1)
   1   FORMAT('+',13X,'<Up>, <Dn>,  <Enter> are active.')

       ELMNTS = 12
       COLUMN = 35

       TPOINT = POINTR

       CALL MENU(ELMNTS,COLUMN,POINTR)

C CLEAR THE BOTTOM LINE OF THE SCREEN.

       CALL CBL

       GO TO(1100,1300,1500,1700,1900,2300,2500,2700,2100,
     * 2900,3100,3300),POINTR
```

```
1100    WRITE(1,5)
   5    FORMAT('+','New NSECT ? (recomputes SCA)  I format  ')
        READ(1,6) NSECT
   6    FORMAT(I2)
        IF(NSECT.GT.10) NSECT = 10
        IF(NSECT.LE.0.0) NSECT = 1
        ISECT = MIN0(ISECT,NSECT)
        SCA = 0.0
        DO 600 J = 1,NSECT
 600    SCA = SCA + SSCA(J)
        GO TO 300

1300    WRITE(1,7)
   7    FORMAT('+New value for grain loading? F format  ')
        READ(1,8) DL
   8    FORMAT(F5.0)
        GO TO 300

1500    WRITE(1,9)
   9    FORMAT('+New value for particle MMD ?  F format  ')
        READ(1,8) PMMD
        GO TO 300

1700    WRITE(1,11)
  11    FORMAT('+New value for TEMPERATURE ? F format  ')
        READ(1,8) TEMP
        GO TO 300

1900    WRITE(1,13)
  13    FORMAT('+New value for SNEAKAGE FACTOR ?  F format  ')
        READ(1,8) SN
        GO TO 300

2100    WRITE(1,15)
  15    FORMAT('+New value for RAPPING ENTRAINMENT ? F format  ')
        READ(1,8) RR(ISECT)
        GO TO 300

2300    WRITE(1,17)
  17    FORMAT('+New SCA ? (recomputes sSCA) F format (0.0 ABRTS) ')
        READ(1,8) TSCA
        IF(TSCA.EQ.0.0) GO TO 300
        SCA = TSCA
        DO 200 J = 1,10
 200    SSCA(J) = TSCA/NSECT
        GO TO 300
```

```
2500    WRITE(1,19)
  19    FORMAT('+New sSCA ? (recomputes SCA) F format (0.0 ABORTS) ')
        READ(1,8) TSCA
        IF(TSCA.EQ.0.0) GO TO 300
        SSCA(ISECT) = TSCA
        SCA = 0.0
        DO 400 J = 1,NSECT
 400    SCA = SCA + SSCA(J)
        GO TO 300

2700    WRITE(1,21)
  21    FORMAT('+New EFIELD value ? (kV/cm) F format  ')
        READ(1,8) EFLD(ISECT)
        GO TO 300

2900    ISECT = ISECT + 1
        IF(ISECT.GT.NSECT) ISECT = 1
        GO TO 300




3100    CONTINUE

C THE RUN TIME PARAMETER MENU IS DISPLAYED.

        POINTR = 9

 500    CALL CLS

        WRITE(1,101) TIME,TAVG,STCKDM,DLDEV,NLNS,ISECT,
     *  RAPT(ISECT),THD(ISECT)
 101    FORMAT('+',10X,'Current RUN-TIME Parameters are:'//
     *  '+',35X,'TIME         =',F8.0,' min'/
     *  '+',35X,'AVGING TIME  =',F8.0,' min'/
     *  '+',35X,'STACK DIAM   =',F8.0,' ft'/
     *  '+',35X,'DL VARIANCE  =',F8.2/
     *  '+',35X,'# Rap Lanes  =',I6,' /sect'/
     *  '+',5X,'For Section # ',I2,14X,'RPPING INTVL =',F8.0,' min'/
     *  '+',35X,'LYR THCKNSS  =',F8.1,' mm'/
     *  '+',35X,'Next Section'/
     *  '+',35X,'ESP Parameters'/
     *  '+',35X,'Return to Main Menu')

        CALL CBL

        WRITE(1,1)
```

```
      ELMNTS = 10
      COLUMN = 33

      TPOINT = POINTR

      CALL MENU(ELMNTS,COLUMN,POINTR)

      CALL CBL

      GO TO(4100,4300,4500,4700,5500,4900,5100,5300,
     *  700,3300),POINTR

4100  WRITE(1,103)
 103  FORMAT('+New point in time ?  F format  ')
      READ(1,8) TIME
      GO TO 500

4300  WRITE(1,105)
 105  FORMAT('+New Averaging Period ?  F format  ')
      READ(1,8) TAVG
      GO TO 500

4500  WRITE(1,113)
 113  FORMAT('+New STACK DIAMETER (ft) ?  F format  ')
      READ(1,8) STCKDM
      GO TO 500

4900  WRITE(1,107)
 107  FORMAT('+New RAPPING INTERVAL ?  F format  ')
      READ(1,8) RAPT(ISECT)
      GO TO 500

5100  WRITE(1,109)
 109  FORMAT('+New Dust Layer THICKNESS ?  F format  ')
      READ(1,8) THD(ISECT)
      GO TO 500

4700  WRITE(1,111)
 111  FORMAT('+New Dust Loading VARIANCE ?  F format  ')
      READ(1,8) DLDEV
      GO TO 500

5300  ISECT = ISECT + 1
      IF(ISECT.GT.NSECT) ISECT = 1
      GO TO 500

5500  WRITE(1,115)
 115  FORMAT('+# lanes for RAPPING DIST ? I format  ')
      READ(1,10) NLNS
  10  FORMAT(I5)
      GO TO 500

3300  RETURN

      END
      SUBROUTINE SAVE

C   PROGRAM TO SAVE THE ESP DATA BLOCK

      LOGICAL FO,FN,TXT(36)
      COMMON /FILDAT/FN(13)
      DATA TXT/'D','U','M','P',' ','F','I','L','N','A','M',
     * 'E','1','/','E','S','P',' ','5','2','0','0','H',' ',
     * '5','3','C','D','H',' ','4','2','0','D','H',X'0D'/

C CLEAR THE SCREEN

      CALL CLS

      WRITE(1,2)
    2 FORMAT('+',19X,'D A T A  S A V E')

      WRITE(1,3) (FN(I),I=1,12)
    3 FORMAT(
     * //'+Enter a filename under which the data is (will be) stored.'
     * /'+The extension must be appended to the name given.'
     * /'+If <ENTER> is pressed, the current name will be used.'
     * //'+The current name is:',1X,12A1/)

      CALL GETNAM

C CHECK FOR EXISTENCE OF A FILE ALREADY

      CALL FCHECK(FO,FN)

      IF(FO.EQ..FALSE.) GO TO 100

      WRITE(1,4)
    4 FORMAT(//'+FILE ALREADY EXISTS.  Press <BREAK> to abort save,'/
     * '+or any other key to continue.')

      IASC = INKEY(1)
      IF(IASC.EQ.1) RETURN

C LOAD THE FILNAME INTO THE "DUMP" ARRAY

100   DO 200 I = 1,12
      TXT(I+5) = FN(I)
200   CONTINUE

      CALL DOSCLL(TXT)

      WRITE(1,7) (TXT(I),I=6,17)
    7 FORMAT(//'+ Data is stored in ',12A1/
     * '+ Hit any key to RETURN')

      IASC = INKEY(1)

      RETURN
      END
```

ESP Dynamic Display Model Listing

```
      SUBROUTINE LOAD

C   PROGRAM TO LOAD THE ESP DATA BLOCK

        LOGICAL FO,FN,TX(18),CR
        COMMON /FILDAT/FN(13)
        DATA TX/'L','O','A','D',' ','F','I','L','N','A','M',
     *  'E','1','/','E','S','P',X'0D'/CR/X'0D'/

C CLEAR THE SCREEN

        CALL CLS

        WRITE(1,2)
   2    FORMAT('+',19X,'D A T A  L O A D')

        WRITE(1,3) (FN(I),I=1,12)
   3    FORMAT(
     *  //'+Enter a filename under which the data is (will be) stored.'
     *  /'+The extension must be appended to the name given.'
     *  /'+If <ENTER> is pressed, the current name will be used.'
     *  //'+The current name is:',1X,12A1/)

        CALL GETNAM

        CALL FCHECK(FO,FN)

        IF(FO.EQ..TRUE.) GO TO 300

        WRITE(1,4)
   4    FORMAT('+',19X,'FILE DOES NOT EXIST'//
     *  '+',19X,'Press any key to continue.')

        IASC = INKEY(1)

        RETURN

 300    DO 400 I = 1,12
        TX(I+5) = FN(I)
        IF(FN(I).EQ.32) TX(I+5) = CR
 400    CONTINUE

        CALL DOSCLL(TX)

        WRITE(1,5) (TX(I),I=6,17)
   5    FORMAT(//'+ Data has been loaded from ',12A1/
     *  '+ Hit any key to RETURN')

        IASC = INKEY(1)

        RETURN

        END
      SUBROUTINE GETNAM

       LOGICAL C,D,FN,BLNK
       COMMON /FILDAT/ FN(13)
       DATA C,D,BLNK/X'1B',X'1E',' '/
       DATA FN/ 'F','I','L','E','N','A','M','E','/', 'E','S','P',' '/

       DO 200 I = 1,12
       IASC = INKEY(1)

C CHECK FOR <ENTER>

       IF(IASC.EQ.13) GO TO 100

       FN(I) = IASC

C CHECK FOR LEFT ARROW KEY TO BACKSPACE

       IF(IASC.EQ.8) I = I - 2

       NUMGCH = I

C "C" IS AN UP LINEFEED CHARACTER, "D" IS ERASE TO END OF LINE

       WRITE(1,10) C,(FN(J),J=1,NUMGCH),D
 10    FORMAT('+'A1,21X,13A1)

 200   CONTINUE

 100   IF(I.EQ.1) RETURN

C BLANK OUT THE REST OF THE FILE NAME.

       NUMGCH = NUMGCH + 1

       DO 400 I = NUMGCH,13
 400   FN(I) = BLNK

       RETURN

       END
      SUBROUTINE PRTOPT

      INTEGER COLUMN,POINTR,ELMNTS,PSEL
      LOGICAL LRAP
      COMMON /PRT/IPRT,LRAP(10),ISEL,PSEL

 100  POINTR = ISEL

C CLEAR THE SCREEN

      CALL CLS

      WRITE(1,2)
   2  FORMAT('+',15X,'Printer Options are:'/
     * '+    Select a strategy or a printer to use'/
     * '+',38X,'Graph MASS and OPACITY'/
     * '+',38X,'Graph THICKNESSES'/
     * '+',38X,'Print NUMERIC VALUES'//
     * '+',38X,'Use Line Printer IV'/
     * '+',38X,'Use Line Printer VIII'/
     * '+',38X,'Use Okidata 84 Printer'/
     * '+',38X,'Use MX 80/Gemini 10'//
     * '+',38X,'Return to Menu')

      CALL CBL

      WRITE(1,1)
   1  FORMAT('+',13X,'<Up>, <Dn>,  <Enter> are active.')

      ELMNTS = 10
      COLUMN = 35

      CALL MENU(ELMNTS,COLUMN,POINTR)

      IF(POINTR.LE.3) ISEL = POINTR
      IF(POINTR.GT.4.AND.POINTR.LE.8) PSEL = POINTR - 4

      RETURN

      END
```

```
        SUBROUTINE RAPIN

         INTEGER COLUMN,POINTR,ELMNTS
         COMMON /RAPS/TIMRAP,THDRAP,CRTTH ,D1,D2

C CLEAR THE SCREEN

         POINTR = 1

 100     CALL CLS

C IDENTIFY THE SUBROUTINE

         WRITE(1,2) TIMRAP,THDRAP,CRTTH
   2     FORMAT('+',20X,'Rapping Schemes'//
     *   '+',2X,'        Options:       ',12X,'Timed Rapping',
     *   ,6X,'<',A4,'>'/
     *   '+',34X,'Thickness Rapping',2X,'<',A4,'>'/
     *   '+',34X,'Critical Thickness =',F5.1,' mm'/
     *   '+',34X,'Return to Menu')

         CALL CBL

         WRITE(1,1)
   1     FORMAT('+',13X,'<Up>, <Dn>,  <Enter> are active.')

         ELMNTS = 4
         COLUMN = 31

         CALL MENU(ELMNTS,COLUMN,POINTR)

         CALL CBL

         GO TO (1100,1300,1500,1700),POINTR

1100     TIMRAP = D1
         THDRAP = D2
         GO TO 100

1300     TIMRAP = D2
         THDRAP = D1
         GO TO 100

1500     WRITE(1,101)
 101     FORMAT('+New CRITICAL THICKNESS (mm) ? F format  ')
         READ(1,102) CRTTH
 102     FORMAT(F5.0)
         GO TO 100

1700     RETURN

         END
```

```
      SUBROUTINE SCRNMS

C THIS ROUTINE WILL DISPLAY TWO SCREEN MESSAGES.

      LOGICAL A(14),ONE,TWO
      DATA A/'L','O','A','D',' ','M','E','S','S','A',
     * 'G','E','1',X'0D'/ONE/'1'/TWO/'2'/

      A(13) = ONE
      CALL DOSCLL(A)

      IASC = INKEY(1)
      IF(IASC.EQ.1) RETURN

      A(13) = TWO
      CALL DOSCLL(A)

      IASC = INKEY(1)

      RETURN
      END
'Z<M       SUBROUTINE PRUNEP

      COMMON /PRT/IPRT

C CLEAR THE SCREEN

      CALL CLS

C SET UP THE PRINTER FOR 132 COLUMNS

      CALL CONPRT

C ACTIVATE PRINTER DURING RUNNING MODEL

      IPRT = 1

C PRINT THE ESP DATA

      CALL PRTHD

C PRINT SYMBOLS AND AXIS.

      CALL PRTGRF

      CALL RUNEP

      RETURN

      END
      SUBROUTINE RUNEP0

       REAL MRAP
       COMMON /RTME/TIME,TAVG,RAPT(10),THD(10),DLDEV,STCKDM,NLNS
       COMMON /RAPI/L,J(10),MRAP(10)
       COMMON /RAVG/Y(240),Z(6)

       DO 200 I = 1,10
       MRAP(I) = 0.0
200    THD(I) = 0.0

       DO 400 I = 1,6
400    Z(I) = 0.0

       DO 600 I = 1,240
600    Y(I) = 0.0

       TIME = 0.0

       CALL RUNEP

       RETURN

       END


      SUBROUTINE RUNEP

       LOGICAL KS,LRAP,LCT(10)
       REAL MMD,MRAP,MRAP0(10)

       DIMENSION JRP(10)

       COMMON /PRT/1PRT,LRAP(10),ISEL
       COMMON /EPPAR/NSECT,DL,PMMD,TEMP,SN,SCA
       COMMON /SECTL/SSCA(10),EFLD(10),RR(10)
       COMMON /RTME/TIME,TAVG,RAPT(10),THD(10),DLDEV,STCKDM,NLNS
       COMMON /RAPS/TIMRAP,THDRAP,CRTTH,D1,D2
       COMMON /RAPI/IRPFLG,IRAP(10),MRAP(10)
       COMMON /DIS/NDIS,NFS,IADR,NPT,OUTDL,OUTBAR,OP,OPBAR
       COMMON /MET/MMD,TDK,SCAM(10),E(10),EPSVIS,OMSN

       DATA OUTBAR,OPBAR,LCT/0.0,0.0,10*0/
       DATA JRP/10*0/

C ZERO THE DAC FOR OPACITY IF IT IS USED.

       CALL OUT(4,0)

C PUT THE CURRENT DISPLAY UP.

       CALL DISPLY(NDIS)
```

ESP Dynamic Display Model Listing

```
C CONVERT THE INPUT DATA TO METRIC UNITS

        CALL CONVRT

C IF TIMED RAPPING IS TO BE USED, SET UP THE RAPPING SCHEDULE.

        DRAP = (FLOAT(NLNS) - 1.0)/FLOAT(NLNS)
        IRPFLG = 0
        IF(TIMRAP.EQ.D1) CALL RPSCHD

C IF PRINTER OUTPUT IS DESIRED, SET PRINT FLAG

        IPFLG = 0
        IF(IPRT.EQ.1) IPFLG = 1
        IPRT = 0

        TMMD = PMMD*1.0E-06

C
 100    TIME = TIME + 1.0

C
C REINITIALISE INLET CONDITIONS FOR EACH PASS.

        TDL = DL
        MMD = TMMD
        LR = 0

C THE DUST LOADING VARIES RANDOMLY, WITH A SHORT PERIOD AND
C A LONG PERIOD.

        IF(DLDEV.EQ.0.0) GO TO 300

        CALL RANDL(U1)

        TDL = DL*(1.0 + DLDEV*U1)

 300    CONTINUE

C SCAN FOR THE LETTER N

        IASC = INKEY(0).AND.95
        IF(IASC.EQ.78) CALL NXTDIS

        DO 1000 KS = 1,NSECT

        RAPDL = 0.0
        P = PEN(KS)
        OUTDL = TDL*P

C SUM THE AMOUNTS COLLECTED FOR LATER RAPPING.

        MRAP(KS) = MRAP(KS) + (1.0 - P)*TDL

C CONVERT COLLECTED MASS TO THICKNESS.
```

```
      THD(KS) = MRAP(KS)/SCAM(KS)*0.66

C IRPFLG=1 IMPLIES TIMED RAPPING

      IF(IRPFLG.EQ.1) GO TO 1200

      IF(THD(KS).LT.CRTTH) GO TO 1400

      LCT(KS) = LCT(KS) + 1
      IF(LCT(KS).EQ.1) MRAP0(KS) = MRAP(KS)
      IF(LCT(KS).GE.NLNS) LCT(KS) = 0
      RAPDL = RR(KS)*MRAP0(KS)/NLNS
      MRAP(KS) = MRAP(KS) - MRAP0(KS)/NLNS

      CALL POKSCT(KS)

      LR = LR + 1
      LRAP(LR) = KS

      GO TO 1400

1200  TTIME = TIME - (IRAP(KS) + JRP(KS))
      IF(AMOD(TTIME,RAPT(KS)).NE.0.0) GO TO 1400

      LCT(KS) = LCT(KS) + 1
      IF(LCT(KS).EQ.1) MRAP0(KS) = MRAP(KS)
      IF(LCT(KS).GE.NLNS) LCT(KS) = 0
      RAPDL = RR(KS)*MRAP0(KS)/NLNS
      MRAP(KS) = MRAP(KS) - MRAP0(KS)/NLNS

      CALL POKSCT(KS)

      LR = LR + 1
      LRAP(LR) = KS

      JRP(KS) = JRP(KS) + 1
      IF(JRP(KS).GE.NLNS) JRP(KS) = 0

1400  TDL = OUTDL + RAPDL
      MMD = (MMD*OUTDL + 6.0E-06*RAPDL)/TDL
      OUTDL = TDL

      IASC = INKEY(0).AND.95
      IF(IASC.EQ.78) CALL NXTDIS

1000  CONTINUE

C CONVERT MASS TO OPACITY.

      CALL CFUNC

C AVERAGE MASS AND OPACITY OVER THE LAST AVERAGING PERIOD.

      CALL RUNAVG
```

```
C DISPLAY THE SELECTED VIDEO MEMORY

        CALL SHIFT

        CALL RTCOL

        CALL PTCOL

        CALL DISPLY(NDIS)

C (P)rint WILL PUT DATA TO THE PRINTER AS LONG AS IT IS HELD.
C TEST FOR (N)ext Display
C SCAN THE KEYBOARD FOR (M)enu , (F)aster, AND (S)lower.

        IASC = INKEY(0).AND.95

        IF(IASC.EQ.80) GO TO 700

        IF(IASC.EQ.78) CALL NXTDIS

        IF(IASC.EQ.77) RETURN

        IF(IASC.EQ.70) NFS = MAX0(NFS/2,1)

        IF(IASC.EQ.83) NFS = MIN0(2*NFS,16380)

        DO 200 K = 1,NFS
        DO 200 KS = 1,5
 200    CONTINUE

        IF(IPFLG.NE.1) GO TO 100

 700    IF(ISEL.EQ.1) CALL PRTOUT
        IF(ISEL.EQ.2) CALL PRTTHD
        IF(ISEL.EQ.3) CALL PRTNUM

        GO TO 100

        END
```

```
       SUBROUTINE RUNSTT

C THIS ROUTINE EVALUATES THE TIME AVERAGED PERFORMANCE OF
C THE ESP, USING THE D-A-R-N APPROACH.

       LOGICAL KS,A(64)
       REAL MMD,MRAP,LF
       COMMON /EPPAR/NSECT,DL,PMMD,TEMP,SN,SCA
       COMMON /DIS/N1,N2,N3,N4,OUTDL,OUTBAR,OP,OPBAR
       COMMON /SECTL/SSCA(10),EFLD(10),RR(10)
       COMMON /RTME/TIME,TAVG,RAPT(10),THD(10),DLDEV,STCKDM,NLNS
       COMMON /RAPI/IRPFLG,IRAP(10),MRAP(10)
       COMMON /MET/MMD,SCAM(10),E(10),EPSVIS,OMSN

C CLEAR THE SCREEN

       CALL CLS

       WRITE(1,3)
   3   FORMAT('+',20X,'Static Model'/
     * '+','Sect',2X,'Dust Load',3X,'Pen'4X,
     * 'Total Pen',1X,'Opacity',2X,'LF',3X,'MMD')

       CALL CONVRT

       DINDL = DL
       TPEN = 1.0

       DO 1000 KS = 1,NSECT

       P = PEN(KS)
       MRAP(KS) = (1.0 - P)*DINDL
       RAPDL = RR(KS)*MRAP(KS)
       OUTDL = DINDL*P
       TDL = OUTDL + RAPDL
       MMD = (MMD*OUTDL + 6.0E-06*RAPDL)/TDL
       OUTDL = TDL
       P = OUTDL/DINDL
       DINDL = OUTDL
       TPEN = TPEN*P

C CALCULATE OPACITY FOR THIS STAGE

       CALL CFUNC

C CALCULATE LOSS FACTOR FOR THIS STAGE

       LF = SN + RR(KS) - SN*RR(KS)

       WRITE(1,2) KS,OUTDL,P,TPEN,OP,LF,MMD
   2   FORMAT('+',I2,2X,F8.4,2X,F8.4,2X,F8.4,3X,F6.0,F5.2
     * ,6PF6.0)

1000   CONTINUE
```

```
      CALL CBL
      WRITE(1,4)
    4 FORMAT('+Press <P> to print these results or ',
     * '<Enter> to return to menu')

  100 IASC = INKEY(1).AND.95

      IF(IASC.EQ.13) RETURN

      IF(IASC.EQ.80) GO TO 300

      GO TO 100

  300 NLINES = NSECT + 4

C SET THE LINE PRINTER FOR 132 COLUMNS.

      CALL CONPRT

      CALL PRTHD

      CALL SCRPRT(NLINES)

      RETURN

      END


     SUBROUTINE NXTDIS

      COMMON /DIS/NDIS,NFS,IADR,NPT,X,XBAR,Y,YBAR

  100 NDIS = MOD((NDIS+4),4) + 1

      CALL DISPLY(NDIS)

      DO 200 I = 1,8000
  200 CONTINUE

      IASC = INKEY(0).AND.95
      IF(IASC.EQ.78) GO TO 100

      RETURN

      END
```

```
       SUBROUTINE PTCOL

        COMMON /DIS/NDIS,NFS,IADR,NPT,OUTDL,OUTBAR,OP,OPBAR

        X = OUTDL/0.05
        NPT = X*40

C IADR HOLDS THE RELATIVE POSITION WITHIN EACH DISPLAY BLOCK

        IADR = X'03FF'
        CALL PTPOK(NPT,IADR)

C PUT IN THE AVERAGE VALUE TOO.

        X = OUTBAR/0.05
        NPT = X*40
        CALL PTPOK(NPT,IADR)

        X = OUTBAR/0.02
        NPT = X*40
        IADR = X'07FF'
        CALL PTPOK(NPT,IADR)

        NPT = (OP/95.)*40
        IADR = X'0BFF'
        CALL PTPOK(NPT,IADR)

        NPT = (OPBAR/50.)*40
        IADR = X'0FFF'
        CALL PTPOK(NPT,IADR)

C OUTPUT AN OPACITY VALUE TO THE DAC

        NOP = (OPBAR/100.)*255
        CALL OUT(4,NOP)

        RETURN

        END
```

```
      FUNCTION PEN(N)

       LOGICAL N
       REAL MMD
       COMMON /MET/MMD,TDK,SCAM(10),E(10),EPSVIS,OMSN
       COMMON /EPPAR/NSECT,DL,PMMD,TEMP,SN,SCA
       DATA PI/3.14159/TWO/2.0E-06/

C THE PENETRATION IS CALCULATED FOR MMD  AND USE TO FIGURE THE
C MEAN DIAMETER.  THIS CORRECTS A PROBLEM OCCURRING WHEN
C A LOW EFFICIENCY STAGE IS USED, viz., A SHIFT IN MMD.

       EF = EPSVIS*(E(N)*E(N))
       W = MMD*EF
       A = OMSN*EXP(-W*SCAM(N))
       B = SN

       PEN = A + B

       PMD = TWO*(1.0 - A) + MMD*A
       MMD = (A*PMD + B*MMD)/PEN

       RETURN
       END
```

```
      SUBROUTINE RPSCHD

C THIS ROUTINE OFFSETS THE SECTIONAL RAPPING INTERVALS WITHIN
C A SHORT TIME PERIOD, AND THEN EFFECTIVELY STARTS THEM
C RUNNING.

      COMMON /EPPAR/NSECT
      COMMON /RAPI/IRPFLG,IRAP(10),X(10)
      COMMON /RTME/TIME,TAVG,RAPT(10),THD(10),DLDEV,STCKDM,NLNS

      RM = 10000.

      DO 400 I = 1,NSECT
      RM = AMIN1(RM,RAPT(I))
400   CONTINUE

      M = RM/NSECT + 0.5

      N = MAX0(1,M)

      DO 200 I = 1,NSECT
200   IRAP(I) = I*N - 1

      IRPFLG = 1

      RETURN
      END
      SUBROUTINE RUNAVG

C W IS GIVEN A SIX POINT AVERAGE, X IS VARIABLE UP TO 240.

        COMMON /RAVG/Y(240),Z(6)
        COMMON /RTME/TIME,TAVG
        COMMON /DIS/N1,N2,N3,N4,X,XBAR,W,WBAR
        DATA Y,Z/240*0.0,6*0.0/

        XTAIL = XBAR

C THE SHIFT ROUTINE MOVES Z(n) to Z(n+1) and Y(n) to Y(n+1).
C THEN NEW VALUES CAN BE INPUT TO Z(1) AND Y(1).
C PUTTING Y AND Z IN A COMMON BLOCK INSURES THEY ARE
C CONTIGUOUS, AND ALLOWS FOR ZEROING IN RUNEP0.

        CALL RSHIFT(Z(5))

        Z(1) = W
        Y(1) = X

        WBAR = 0.0

        DO 200 I = 1,6
 200    WBAR = WBAR + Z(I)

        WBAR = WBAR/6.0

        XBAR = 0.0

        NMAX = MIN1(TAVG,240.)

        DO 400 I = 1,NMAX
 400    XBAR = XBAR + Y(I)

        XBAR = XBAR/NMAX

        IF(TAVG.LE.240.0) RETURN

C IF THE AVGERAGING PERIOD IS LONGER THAN 240 MINUTES,
C COMPUTE AN EXPONENTIAL AVERAGE OF THE RUNNING AVERAGES

        AVGT = TAVG - 240.
        AVGT1 = AVGT - 1.0
        XBAR = (XBAR + XTAIL*AVGT1)/AVGT

        RETURN

        END
```

```
       SUBROUTINE CFUNC

        REAL KGPCM,MMD
        COMMON /RTME/A,B,C(10),D(10),E,STCKDM,NLNS
        COMMON /MET/MMD
        COMMON /DIS/N1,N2,N3,N4,OUTDL,OUTBAR,OP,OPBAR
        DATA FORHPI,GRTOGM,HMD,QFAC/1256.6,0.00229, 0.7E-06,1.57/

C CALCULATE THE AVERAGE NUMBER OF PARTICLES PER CUBIC METER
C FROM THE OUTDL.

        KGPCM = OUTDL*GRTOGM
        GMD = MMD

C PARTICLE MASS IS 4*PI*RAD**3*DENSITY (2400 KG/M3).

        PARTMS = FORHPI*(GMD**3)
        PARTNM = KGPCM/PARTMS

C THE QEXT FACTOR IS 2.0 TIMES THE PROJECTED PARTICLE AREA.

        QEXT = QFAC*(GMD*GMD)
        BEXT = PARTNM*QEXT
        ABSOR = BEXT*STCKDM*0.305
        IF(ABSOR.LT.0.2) GO TO 100

        TRANS = EXP(-ABSOR)
        OP = 100.*(1.0 - TRANS)

        RETURN

 100    OP = 100.*ABSOR

        RETURN

        END
       .
       .
       .

      SUBROUTINE CONVRT

       LOGICAL K
       REAL MMD
       COMMON /EPPAR/NSECT,DL,PMMD,TEMP,SN,SCA
       COMMON /SECTL/SSCA(10),EFLD(10),RR(10)
       COMMON /MET/MMD,TDK,SCAM(10),E(10),EPSVIS,OMSN
       DATA EPSO/8.845E-12/

       MMD = PMMD*1.0E-06
       TDK = 5.0*(TEMP + 459.)/9.0

       OMSN = 1.0 - SN

       DO 200 K = 1,10
       SCAM(K) = SSCA(K)/5.079/OMSN
       E(K) = EFLD(K)*1.0E05
 200   CONTINUE

       VIS = 1.72E-05*(TDK/273.)**0.71
       EPSVIS = EPSO/VIS

       RETURN

       END



      SUBROUTINE RANDL(U1)

       REAL LT
       DATA ST,LT/0.0,0.0/

C THIS ROUTINE CALLS RAN REPEATEDLY AND PERFORMS LONG AND
C SHORT TERM EXPONENTIAL AVERAGES, WHICH HAVE THE EFFECT
C OF MAKING THE DISTRIBUTION NORMAL.

       U1 = RAN(1.0)

C CONVERT THE VARIABLE ON (0,1) TO (-1,1)

       U1 = 2.0*U1 - 1.0

       LT = (119.*LT + U1)/120.0
       ST = (19.*ST  + U1)/20.0

       U1 = (ST + LT)/2.0*11.9

C THE FACTOR OF 11.9 COMES FROM SQRT(120+20)

       RETURN

       END
      SUBROUTINE PRTHD

       LOGICAL P1,P2
       COMMON /EPPAR/NSECT,DL,PMMD.TEMP,SN,SCA
       COMMON /SECTL/SSCA(10),EFLD(10),RR(10)
       COMMON /RTME/TIME,TAVG,RAPT(10),THD(10),DLDEV, STCKDM,NLNS
       COMMON /RAPI/IRPFLG
       COMMON /RAPS/TIMRAP,THDRAP,CRTTH,D1,D2

C SELECT NORMAL LINE SPACING

       CALL NRMLF(P1,P2)

       IRPFLG = 0
       IF(TIMRAP.EQ.D1) IRPFLG = 1

       WRITE(2,1)
  1    FORMAT('+',30X,'DATA FOR THE REAL TIME RUNNING MODEL')

       WRITE(2,2)
  2    FORMAT(' ',35X,'Precipitator Parameters:')

       WRITE(2,3) NSECT,SCA,SN,NLNS,TEMP
  3    FORMAT('0','Num Sect = ',I2,2X,'Total SCA = ',
     * F5.0,' ft2/kACFM',2X,'Sneakage = ',F3.2,' /sect',
     * 2X,'Rap Lanes = ',I2,' /sec',2X,'Temp = '
     * ,F5.0,' Deg F.')

       WRITE(2,4) (SSCA(I),I=1,NSECT)
  4    FORMAT(' ',15X,'Sectional SCA''s = ',10F7.1)

       WRITE(2,5) (EFLD(I),I=1,NSECT)
  5    FORMAT(' ',10X,'Sect Fields (kV/cm)  = ',10F7.2)

       WRITE(2,10) (RR(I),I=1,NSECT)
 10    FORMAT(' ',19X,'Rap Factors = ',10F7.2)

       IF(IRPFLG.EQ.1) WRITE(2,6) (RAPT(I),I=1;NSECT)
  6    FORMAT(' ',16X,'Rapper Periods = ',10F7.0)

       IF(IRPFLG.NE.1) WRITE(2,9) CRTTH
  9    FORMAT(' ',19X,'Thickness Rapping is active.',5X,
     * 'The critical thickness = ',F5.2,' mm')

       WRITE(2,8) (THD(I),I=1,NSECT)
  8    FORMAT(' ',14X,'Layer Thickness  = ',10F7.2)

       WRITE(2,7) DL,DLDEV,PMMD,STCKDM,TIME.TAVG
  7    FORMAT(' ',2X,'Dust Load = '.F4.1,' gr/ACF',2X,'DL VAR = ',
     * F3.2,2X,'Part MMD = ',F4.1,2X,'Stack Diam = ',F3.0,
     * ' ft',2X,'Time = ',F5.0,' min',2X,'Avging Per = '
     * ,F4.0,' min')

       RETURN
       END
      SUBROUTINE PRTGRF

       INTEGER PSEL
       LOGICAL B,A,B3,B4,B5,B6,LRAP,I,P1,P2
       COMMON /PRT/IPRT,LRAP(10),ISEL,PSEL
       COMMON /SCPRT/B3,B4,B5,B6,B(121),A(9)
       COMMON /PRTR/P1,P2
       DATA I/'I'/

       CALL NRMLF(P1,P2)

       DO 200 J = 1,121
200    B(J) = A(1)

       WRITE(2,1) A,B
  1    FORMAT('+',1X,130A1)

       IF(ISEL.EQ.1) WRITE(2,2)
  2    FORMAT('+',' Time',' Rap ',' Mass (m)',5X,'Mass Avg (+)',
    *  5X,'Opacity (o)',5X,'Opacity Avg (:)'/
    *  '+',10X,'0 gr/acf',21X,'.02',27X,'.04',27X,
    *  '.06',26X,'.08'/'+',10X,'0 % Opacity',19X,'25 %',26X,
    *  '50 %',26X,'75 %',22X,'100 %')

       IF(ISEL.EQ.2) WRITE(2,4)
  4    FORMAT('+',10X,'Layer Thicknesses (mm) by Section numbers'/
    *  '+',10X,'0 mm',26X,'1 mm',26X,'2 mm',26X,
    *  '3 mm',23X,'4 mm')

       IF(ISEL.GT.2) GO TO 100

C PERFORM A ZERO LINE FEED FOR CERTAIN PRINTERS

       CALL ZERLF(P1,P2)

       WRITE(2,1) A,B

C PERFORM A REVERSE LINE FEED FOR STRIKING OVER

       CALL REVLF(P1,P2)

C WRITE THE AXIS MARKERS OVER THE PREVIOUS LINE AND SET
C UP THE HALF-LINEFEED REQUIREMENTS

       WRITE(2,5) I,I,I,I,I
  5    FORMAT('+',10X,A1,29X,A1,29X,A1,29X,A1,29X,A1)

       CALL HLFLF(P1,P2)

       RETURN
```

```
100    WRITE(2,7)
  7    FORMAT('+',' Time',2X,'Last Rap',5X,'Outlet DL ',
     * ' Out Avg  ',' Opacity  ',' Opac Avg',10X, 'Thickness',
     * ' in mm')

       RETURN

       END


      SUBROUTINE PRTTHD

       LOGICAL B,B3,B4,B5,B6,SYMB(10),BLNK,I,DOT,LRAP,R(10)
       LOGICAL P1,P2,CHR
       DIMENSION IQ(10)
       COMMON /PRT/IPRT,LRAP(10)
       COMMON /RTME/TIME,TAVG,RAPT(10),THD(10)
       COMMON /EPPAR/NSECT
       COMMON /SCPRT/B3,B4,B5,B6,B(121)
       COMMON /PRTR/P1,P2

       DATA BLNK,I,DOT/' ','I','.'/
       DATA SYMB/'1','2','3','4','5','6','7','8','9'.'0'/
       DATA IBS/0/

       IBS = MOD((IBS+2),2) + 1
       IT = AMOD(TIME,20.)

C BLANK THE ARRAY OR FILL IT WITH DOTS EVERY TENTH MINUTE

       CHR = BLNK
       IF(IT.EQ.0) CHR = DOT

       DO 200 K = 1,119,2
       B(K+1) = BLNK
200    B(K) = CHR

C PUT IN THE AXIS MARKERS

       B(1) = I
       B(31) = DOT
       B(61) = DOT
       B(91) = DOT
       B(121) = DOT

C FIGURE THE POSITIONS OF THE QUANTITIES AND PUT IN SYMBOLS

       DO 400 K = 1,NSECT
       IQ(K) = THD(K)/4.0*120 + 1
       IQ(K) = MIN0(IQ(K),121)
       M = MAX0(IQ(K),0)
       B(M) = SYMB(K)
400    CONTINUE
```

```
C FIND THE MAXIMUM PRINT POSITION EXCEPT WHEN PRINTING DOTS

        IMAX = 0
        IF(IT.EQ.0) IMAX = 121

        DO 600 K = 1,NSECT
        IMAX = MAX0(IMAX,IQ(K))
 600    CONTINUE

        IF(IMAX.GT.91) IMAX = 121
        IF(IMAX.LT.91.AND.IMAX.GT.61) IMAX = 91
        IF(IMAX.LT.61.AND.IMAX.GT.31) IMAX = 61
        IF(IMAX.LT.31) IMAX = 31

C FIND OUT THE LAST SECTIONS RAPPED

        DO 800 K = 1,10
 800    R(K) = BLNK

        K = 0
        DO 1000 IR = 1,10
        J = 11 - IR
        IF(LRAP(J).EQ.0) GO TO 1000
        IF(LRAP(J).EQ.10) LRAP(J) = 0
        K = K + 1
        R(K) = LRAP(J) + X'30'
        LRAP(J) = 0
1000    CONTINUE

        IT = TIME
        IPRT = 2

C SKIP THE TIME PRINT IF IBS = 0.

        IF(IBS.EQ.1) GO TO 500

        WRITE(2,3) P1,P2,IT,R(1),R(2),R(3),R(4),(B(K),K=1,IMAX)
   3    FORMAT('+',2A1,I5,1X,4A1,121A1)

        RETURN

 500    WRITE(2,4) P1,P2,R(1),R(2),R(3),R(4),(B(K),K=1,IMAX)
   4    FORMAT('+',2A1,6X,4A1,121A1)

        RETURN

        END



        SUBROUTINE PRTOUT

        LOGICAL B,BLNK,STAR,PLUS,I,K,
     *  CLN,EXC,DOT,LRAP,R(10),B3,B4,B5,B6
        LOGICAL P1,P2,CHR
        COMMON /DIS/NDIS,NFS,IADR,NPT,OUTDL,OUTBAR,OP,OPBAR
        COMMON /RTME/TIME
        COMMON /PRT/IPRT,LRAP(10)
        COMMON /SCPRT/B3,B4,B5,B6.B(121)
        COMMON /PRTR/P1,P2
        DATA BLNK,STAR,PLUS,I/' ','m','+','I'/
        DATA CLN,EXC,DOT/':','o','.'/
        DATA IBS/0/

        IBS = MOD((IBS+2),2) + 1
        IT = AMOD(TIME,20.)

C BLANK THE ARRAY OR FILL IT WITH DOTS EVERY TENTH MINUTE

        CHR = BLNK
        IF(IT.EQ.0) CHR = DOT

        DO 200 K = 1,119,2
        B(K+1) = BLNK
 200    B(K) = CHR

C PUT IN THE AXIS MARKERS

        B(1) = I
        B(31) = DOT
        B(61) = DOT
        B(91) = DOT
        B(121) = DOT

C FIGURE THE POSITIONS OF THE QUANTITIES

        IM = OUTDL/0.08*120 + 1
        IM = MIN0(IM,121)
        IMB = OUTBAR/0.08*120 + 1
        IMB = MIN0(IMB,121)
        IO = OP/100.*120 + 1
        IO = MIN0(IO,121)
        IOB = OPBAR/100.*120 + 1
        IOB = MIN0(IOB,121)

C PUT IN THE SYMBOLS FOR THE QUANTITIES

        B(IOB) = CLN
        B(IO) = EXC
        B(IMB) = PLUS
        B(IM) = STAR

C FIND THE MAXIMUM POSITION TO BE PRINTED, EXCEPT WHEN
C PRINTING DOTS

        IMAX = MAX0(IM,IMB,IO,IOB)
        IF(IT.EQ.0) IMAX = 121
        IF(IMAX.GT.91) IMAX = 121
        IF(IMAX.LT.91.AND.IMAX.GT.61) IMAX = 91
        IF(IMAX.LT.61.AND.IMAX.GT.31) IMAX = 61
        IF(IMAX.LT.31) IMAX = 31

C FIND OUT THE LAST SECTIONS RAPPED

        K = 0
        DO 600 IR = 1,10
 600    R(IR) = BLNK

        DO 400 IR = 1,10
        J = 11 - IR
        IF(LRAP(J).EQ.0) GO TO 400
        IF(LRAP(J).EQ.10) LRAP(J) = 0
        K = K + 1
        R(K) = LRAP(J) + X'30'
        LRAP(J) = 0
 400    CONTINUE

        IT = TIME
        IPRT = 2

C WRITE THE LINE WITH NO TIME IF IBS=1

        IF(IBS.EQ.1) GO TO 300

        WRITE(2,3) P1,P2,IT,R(1),R(2),R(3),R(4),(B(K),K=1,IMAX)
   3    FORMAT('+',2A1,I5,1X,4A1,121A1)

        RETURN

 300    WRITE(2,4) P1,P2,R(1),R(2),R(3),R(4),(B(K),K=1,IMAX)
   4    FORMAT('+',2A1,6X,4A1,121A1)

        RETURN
        END
```

```
      SUBROUTINE PRTNUM

       INTEGER PSEL
       LOGICAL LRAP,R(10),K,BLNK
       COMMON /PRT/IPRT,LRAP(10),ISEL,PSEL
       COMMON /EPPAR/NSECT
       COMMON /RTME/TIME,TAVG,RAPT(10),THD(10)
       COMMON /DIS/N1,N2,N3,N4,OUTDL,OUTBAR,OP,OPBAR
       DATA BLNK/' '/

C FIND OUT THE LAST SECTIONS RAPPED.

       DO 200 K = 1.10
 200   R(K) = BLNK

       K = 0
       DO 400 IR = 1,10
       J = 11 - IR
       IF(LRAP(J).EQ.0) GO TO 400
       IF(LRAP(J).EQ.10) LRAP(J) = 0
       K = K + 1
       R(K) = LRAP(J) + X'30'
       LRAP(J) = 0
 400   CONTINUE

C PRINT THE DATA

       IT = TIME
       WRITE(2,1)IT,R,OUTDL,OUTBAR,OP,OPBAR,(THD(K), K=1,NSECT)
   1   FORMAT('+',I6,2X,10A1,F8.4,2X,F10.4,F10.1, F10.1,10F7.2)

       RETURN

       END
      SUBROUTINE HLFLF(P1,P2)

      INTEGER PSEL
      LOGICAL LRAP,B3,B4,B5,B6,NULL,P1,P2
      LOGICAL ESC,FS,PCT,NIN,THR,ELEV,CR
      COMMON /SCPRT/B3,B4,B5,B6
      COMMON /PRT/IPRT,LRAP(10),ISEL,PSEL
      DATA NULL/X'01'/ESC/X'1B'/FS/X'1C'/PCT/'%'/NIN/'9'/
      DATA THR/'3'/ELEV/11/CR/X'0D'/

      IF(PSEL.EQ.1) GO TO 100

      P1 = NULL
      P2 = NULL
      IF(PSEL.EQ.2) WRITE(2,1) ESC,FS,CR,CR
      IF(PSEL.EQ.3) WRITE(2,1) ESC,PCT,NIN,ELEV
      IF(PSEL.EQ.4) WRITE(2,1) ESC,THR,ELEV

    1 FORMAT('+',4A1)

      RETURN

  100 P1 = B3
      P2 = B6
      RETURN
      END


      SUBROUTINE NRMLF(P1,P2)

      INTEGER PSEL
      LOGICAL LRAP,B3,B4,B5,B6,NULL,P1,P2
      LOGICAL ESC,TWO,SIX
      COMMON /SCPRT/B3,B4,B5,B6
      COMMON /PRT/IPRT,LRAP(10),ISEL,PSEL
      DATA NULL/X'01'/ESC/X'1B'/TWO/'2'/SIX/'6'/

      P1 = NULL
      P2 = NULL

      IF(PSEL.EQ.2) WRITE(2,1) ESC,SIX
      IF(PSEL.EQ.3) WRITE(2,1) ESC,SIX
      IF(PSEL.EQ.4) WRITE(2,1) ESC,TWO

    1 FORMAT('+',2A1)

      RETURN

      END
      SUBROUTINE REVLF(P1,P2)

       INTEGER PSEL
       LOGICAL LRAP,B3,B4,B5,B6,NULL,P1,P2
       LOGICAL ESC,LF,SIX
       COMMON /SCPRT/B3,B4,B5,B6
       COMMON /PRT/IPRT,LRAP(10),ISEL,PSEL
       DATA NULL/X'01'/ESC/X'1B'/LF/X'0A'/SIX/'6'/

C REVERSE FEEDS ARE NOT POSSIBLE ON THE OKIDATA OR MX80 PRINTERS

       P1 = NULL
       P2 = NULL

       IF(PSEL.LE.2) WRITE(2,1) ESC,LF
   1   FORMAT('+',2A1)

       RETURN

       END



       SUBROUTINE ZERLF(P1,P2)

       INTEGER PSEL
       LOGICAL LRAP,B3,B4,B5,B6,NULL,P1,P2
       LOGICAL ESC,THR
       COMMON /SCPRT/B3,B4,B5,B6
       COMMON /PRT/IPRT,LRAP(10),ISEL,PSEL
       DATA NULL/X'01'/ESC/X'1B'/PCT/'%'/THR/'3'/NIN/'9'/

C ZERO LINE FEEDS ARE NOT POSSIBLE ON THE LPIV OR LPVIII PRINTERS
C ON THE OTHERS, IT IS ACTUALLY 1/144" FORWARD.

       P1 = NULL
       P2 = NULL

       IF(PSEL.EQ.3) WRITE(2,1) ESC,PCT,NIN,NULL
       IF(PSEL.EQ.4) WRITE(2,1) ESC,THR,NULL
   1   FORMAT('+',4A1)

       RETURN

       END
      SUBROUTINE SCRPRT(NLINES)

      LOGICAL A(64)

      CALL CONPRT

C PEEK THE SCREEN AND PRINT IT

      DO 200 I = 1,NLINES
      DO 400 J = 1,64
      IADR = X'3BFF' + 64*(I-1) + J
 400  A(J) = PEEK(IADR)

      WRITE(2,5) A
   5  FORMAT('+',22X,64A1)

 200  CONTINUE

      WRITE(2,6)
   6  FORMAT(' ')

      RETURN

      END

C

      SUBROUTINE CONPRT

      INTEGER PSEL
      LOGICAL B3,B4,B5,B6,LRAP,P1,P2
      COMMON /PRT/IPRT,LRAP(10),ISEL,PSEL
      COMMON /SCPRT/B3,B4.B5.B6
      DATA P1,P2/X'1C',X'0F'/

C SET UP THE PRINTER FOR 132 COLUMNS

      IF(PSEL.LE.2) WRITE(2,3) B3,B5
      IF(PSEL.EQ.3) WRITE(2,3) P1
      IF(PSEL.EQ.4) WRITE(2,3) P2
   3  FORMAT('+',2A1)

      RETURN

      END
```

```
****************** ASSEMBLY LANGUAGE ROUTINES *****************

This section contains the machine language routines for
;displaying the calculations.
;
        ENTRY DISPLY,SHIFT,FILL,RTCOL,PTPOK,TITLES,POKSCT
        ENTRY RSHIFT
        EXTRN DSTART
;
.PRINTX -RSHIFT-
;
;This is the shift  routine for the running average calculation
;On entry, HL points to the next to last value.  By pointing
;DE to the last value, everything can be shifted up one place.
;
RSHIFT:
        PUSH HL
        POP DE                  ;get it into DE
        INC DE
        INC DE
        INC DE
        INC DE                  ;next 4 byte position.
        LD BC,980               ;Y+Z=246 total 4 byte positions
        LDDR                    ;move them downwards
        RET


.PRINTX -DISPLY-
;
;This routine test the contents of HL and jumps to the
;appropriate display routine.
;
DISPLY:
        LD A,1       ;FIRST COMPARISON
        CP (HL)
        JR Z,DSPLY1 ;SHORT JUMP, SO JR
;
        LD A,2       ;SECOND COMPARISON
        CP (HL)
        JR Z,DSPLY2
;
        LD A,3       ;THIRD COMPARISON
        CP (HL)
        JR Z,DSPLY3
;
        LD A,4       ;FOURTH COMPARISON
        CP (HL)
        JR Z,DSPLY4
;
;
```

```
.PRINTX -DSPLYn-
;MOVES VIRTUAL MEMORY TO SCREEN FROM 4K BYTES BELOW THE HIMEM
;
;
DSPLY1: LD BC,000H       ;no offset
        JR MOVE
;
DSPLY2: LD BC,400H       ;1k offset
        JR MOVE
;
DSPLY3: LD BC,800H       ;2k offset
        JR MOVE
;
DSPLY4: LD BC,0C00H       ;3k offset
;
MOVE:   LD HL,(DSTART)   ;start of display block
        ADD HL,BC        ;get correct block
        LD DE,3C00H      ;VIDEO MEMORY
        LD BC,400H       ;1K block move
        LDIR             ;move it
        RET



.PRINTX -SHIFT-
;This routine shifts all the virtual video addresses one to ;the left.
;
SHIFT:
        LD HL,(DSTART)   ;
        LD DE,41H        ;FIRST LOCATION TO BE SHIFTED
        ADD HL,DE
        PUSH HL
        POP DE
        DEC DE           ;FIRST POSITION,LINE 2
        LD BC,37FH       ;LINES 2-15
        LDIR
;
;
        LD HL,(DSTART)   ;
        LD DE,441H       ;FIRST LOCATION TO BE SHIFTED
        ADD HL,DE
        PUSH HL
        POP DE
        DEC DE           ;FIRST POSITION,LINE 2
        LD BC,37FH       ;LINES 2-15
        LDIR
;
```

```
        LD HL,(DSTART)  ;
        LD DE,841H      ;FIRST LOCATION TO BE SHIFTED
        ADD HL,DE
        PUSH HL
        POP DE
        DEC DE          ;FIRST POSITION,LINE 2
        LD BC,37FH      ;LINES 2-15
        LDIR
;
;
        LD HL,(DSTART)  ;
        LD DE,0C41H     ;FIRST LOCATION TO BE SHIFTED
        ADD HL,DE
        PUSH HL
        POP DE
        DEC DE          ;FIRST POSITION,LINE 2
        LD BC,37FH      ;LINES 2-15
        LDIR
;
        RET


.PRINTX -RTCOL-
;This routine loads upper and lower boundary characters into
;appropriate locations and blanks all the other lines in the
;column
;
RTCOL:
        LD HL,(DSTART)  ;virtual video start
        LD BC,007FH     ;LAST POSITION,2ND LINE
        ADD HL,BC
        LD (HL),81H     ;TOP ROW CHARACTER
        LD DE,400H      ;1K INCREMENT
        ADD HL,DE
        LD (HL),81H
        ADD HL,DE
        LD (HL),81H
        ADD HL,DE
        LD (HL),81H
;
        LD HL,(DSTART)
        LD BC,03BFH     ;LAST POSITION,15TH LINE
        ADD HL,BC
        LD (HL),90H     ;BOTTOM ROW CHAR
        ADD HL,DE
        LD (HL),90H
        ADD HL,DE
        LD (HL),90H
        ADD HL,DE
        LD (HL),90H
```

```
;PUT IN THE BLANKS
;
        LD B,12         ;12 LINES TO BE BLANKED
        LD HL,(DSTART)
        LD DE,007FH     ;LAST POSITION, 2ND LINE
        ADD HL,DE
        LD DE,40H       ;INC OF 64
LOOP1:  ADD HL,DE
        LD (HL),80H     ;BLANK GRAPHIC
        DJNZ LOOP1
;
        LD B,12
        LD HL,(DSTART)
        LD DE,047FH     ;PAGE2,LAST POS.2ND LINE
        ADD HL,DE
        LD DE,40H
LOOP2:  ADD HL,DE
        LD (HL),80H
        DJNZ LOOP2
;
        LD B,12         ;12 LINES TO BE BLANKED
        LD HL,(DSTART)
        LD DE,087FH     ;PAGE3,LAST POS,2ND LINE
        ADD HL,DE
        LD DE,40H       ;INC OF 64
LOOP3:  ADD HL,DE
        LD (HL),80H     ;BLANK GRAPHIC
        DJNZ LOOP3
;
        LD B,12
        LD HL,(DSTART)
        LD DE,0C7FH     ;PAGE4,LAST POS,2ND LINE
        ADD HL,DE
        LD DE,40H
LOOP4:  ADD HL,DE
        LD (HL),80H
        DJNZ LOOP4
;
        RET
;
;
.PRINTX -FILL-
;This routine fills the virtual 4k video memory with 80H
;to blank the viewing field.
;
FILL:
        LD HL,(DSTART)  ;FIRST POSITION IN MEM
        LD (HL),80H     ;BLANK CHARACTER
        PUSH HL
        POP DE
        INC DE          ;NEXT POSITION
        LD BC,0FFFH     ;4095 POSITIONS
        LDIR
;
        RET
```

ESP Dynamic Display Model Listing

```
      .PRINTX -TITLES-
        ;This routine loads titles into the high memory virtual video
      ;locations
        ;
        HD1:    DEFM '1 MASS EMISSIONS (0 -.05 gr/acf)                       <Rapped
  0 >'
        HD2:    DEFM '2 AVERAGED EMISSIONS  (0 - .02 gr/acf)                 <Rapped
  0 >'
        HD3:    DEFM '3 OPACITY (0 - 95%)                                    <Rapped
  0 >'
        HD4:    DEFM '4 AVERAGED OPACITY (0 - 50%)                           <Rapped
  0 >'
        BT1:    DEFM 'Press:  (M)enu  (F)aster  (S)lower  (N)ext Display  (P)rin
t    '
        ;
        ;THESE MESSAGES ARE 63 BYTES LONG.  PUT THEM IN THEIR
        ;PROPER LOCATIONS
        ;
        TITLES:
                LD HL,(DSTART)
                LD BC,0000H
                ADD HL,BC
                PUSH HL
                POP DE
                LD HL,HD1
                LD BC,63
                LDIR
        ;
                LD HL,(DSTART)
                LD BC,0400H
                ADD HL,BC
                PUSH HL
                POP DE
                LD HL,HD2
                LD BC,63
                LDIR
        ;
                LD HL,(DSTART)
                LD BC,0800H
                ADD HL,BC
                PUSH HL
                POP DE
                LD HL,HD3
                LD BC,63
                LDIR
        ;
                LD HL,(DSTART)
                LD BC,0C00H
                ADD HL,BC
                PUSH HL
                POP DE
                LD HL,HD4
                LD BC,63
                LDIR
```

```
;
;Now for the bottom line of the display
;
        LD HL,(DSTART)
        LD BC,03C0H
        ADD HL,BC
        PUSH HL
        POP DE
        LD HL,BT1
        LD BC,63
        LDIR
;
        LD HL,(DSTART)
        LD BC,07C0H
        ADD HL,BC
        PUSH HL
        POP DE
        LD HL,BT1
        LD BC,63
        LDIR
;
        LD HL,(DSTART)
        LD BC,0BC0H
        ADD HL,BC
        PUSH HL
        POP DE
        LD HL,BT1
        LD BC,63
        LDIR
;
        LD HL,(DSTART)
        LD BC,0FC0H
        ADD HL,BC
        PUSH HL
        POP DE
        LD HL,BT1
        LD BC,63
        LDIR
;
        RET
```

```
.PRINTX -POKSCT-
;This routine looks at the byte in HL, adds 30H
;to it and displays it in the rapper part of the
;title on each virtual video page.
;
POKSCT:
        LD A,(HL)         ;INSERT TRANSFERRED VALUE
        CP 0AH           ;TEST FOR "10"
        JR NZ,POKR
        XOR A            ;ZERO THE CONTENTS OF A
;
POKR:   ADD A,30H        ;ADD SECTION #
        LD HL,(DSTART)
        LD BC,003CH      ;FIRST LOCATION
        ADD HL,BC
        LD (HL),A
        LD BC,400H       ;1K steps
        ADD HL,BC
        LD (HL),A
        ADD HL,BC
        LD (HL),A
        ADD HL,BC
        LD (HL),A
;
        RET


.PRINTX -PTPOK-
;This rouutine finds the proper row for the value to be
;displayed, and sets the proper bit.
;
;
;(HL) IS NPT AND (DE) IS IADR (the offset within each
;frame ) FROM THE CALLING PROGRAM
;
PTPOK:
         PUSH HL          ;STORE THE POINTER TO NPT
         PUSH DE          ;STORE POINTER TO IADR
         POP HL           ;PUT IADR POINTER IN HL
       LD C,(HL)        ;PUT LOW BYTE OF IADR IN C REG.
         INC HL
         LD B,(HL)        ;HIGH BYTE IN B REG.
;
         LD HL,(DSTART)  ;start of virtual video
         ADD HL,BC
       PUSH HL
         POP BC
```

```
; BC NOW CONTAINS THE VALUE OF IADR WHERE THE DATA IS TO BE
;STORED.  NOW  SET AN UPPER LIMIT ON NPT
;
        POP HL          ;RESTORE POINTER TO NPT.
        LD A,(HL)       ;PUT NPT IN A REGISTER
        CP 41           ;41 IS THE MAXIMUM ALLOWED
        JP M,OK         ;JP IF NEGATIVE
        LD A,41         ;NPT IS GREATER THAN 41
;
OK:                     ;A IS 41 OR LESS AT THIS POINT.
        PUSH BC         ;STORE IADR
        POP HL          ;PUT IADR INTO HL
        LD DE,64        ;INCREMENT FOR ROW ADDRESS
;
        OR A            ;RESET THE SIGN FLAG
INT:    SBC HL,DE       ;SHIFT UP ONE ROW AT LEAST
        SUB 3           ;DECREASE A IN STEPS OF 3
        JP P,INT        ;CONTINUE UNTIL NEGATIVE
                        ;THIS IS THE NEGATIVE REMAINDER
        ADD A,4         ;MAKE IT POSITIVE
;
;COMPARE THE RESULT WITH CERTAIN VALUES(1,2,3) AND
;SET THE APPROPRIATE BITS IN (HL)
;
        CP 1            ;THIS IS THE BIT 5 COMPARE.
        JR NZ,CP2
        SET 5,(HL)
        RET
;
CP2:    CP 2            ;THIS IS BIT 3 COMPARE
        JR NZ,CP3
        SET 3,(HL)
        RET
;
CP3:    CP 3            ;THIS IS BIT 1 COMPARE.
        RET NZ          ;RETURN WITHOUT DOING ANYTHING
        SET 1,(HL)
        RET
;
;
;
```

ESP Dynamic Display Model Listing

```
        ENTRY INKEY,$INIT,MENU,DSTART,CLS,CBL,FCHECK
        ENTRY DOSCLL
        EXTRN $EC,$CLSFL
;
; reserve space for all the variables that will be needed
;
DSTART: DEFS 2
CURPOS: DEFS 2
POINTR: DEFS 2
COLUMN: DEFS 2
ELMNTS: DEFS 2
CHRPTR: DEFS 1
FCB:    DEFS 32
FRATE:  DEFS 1
;
.PRINTX-INKEY-
;
; This routine uses the ROM keyboard routines to perform the
; INKEY function.  If the argument is less than 0, then the
; whole keyboard is addressed at once. If the argument is 0,
; the keyboard is scanned one time only, and the ASCII value
; is returned. If the argument is greater than zero, the keyboard
; is scanned repeatedly until a key is pressed, and its ASCII
; value is returned.
;
; HL  contains the address of the argument upon entry.
;
INKEY:
;
        CALL CLBUFF         ;clear the row history buffer
        LD A,(HL)           ;put in A for comparison
        CP 00h              ;test for - 0 +
        JP M,FSCAN          ;Jump to F(ast) SCAN
        JP Z,ZSCAN          ; Z(ero) SCAN
;
        CALL 0049H          ;scan until key pressed.
        LD L,A              ;transfer A to INKEY via
        LD H,00H            ;HL register
        RET
;
ZSCAN:                      ;scan one time only
        CALL 002BH          ;and return value in A
        LD L,A
        LD H,00H
        RET
;
FSCAN:                      ;scan all rows at once
        LD HL,38FFH         ;including shift key
        LD A,(HL)
        LD L,A
        LD H,00H
        RET
;
;
;
```

```
;        CLBUFF
;
;This routine clears the row history buffer.
;
CLBUFF:
        XOR A           ;zero A
        LD B,7          ;there are seven buffers to zero
        LD DE,4036H     ;starting here.
BUFF:   LD (DE),A       ;clear row history buffer entry
        INC DE          ;put next address
        DJNZ BUFF       ;do it seven times.
        RET



.PRINTX-CLS-
;
;This is the ROM CLear Screen routine, the same as writing -1C,1F- to
;the keyboard unit number.
;
CLS:    JP 01C9H
;
.PRINTX-CBL-
;
;This is the Clear Bottom Line routine, positioning the cursor to
;overwrite the bottom line on the screen.
;
CBL:    LD HL,3F80H             ;desired cursor position
        LD (4020H),HL           ;cursor storage location
        RET



.PRINTX-$INIT-
;
;This routine overides the FORLIB $INIT routine, so that the high
;memory pointer is properly set; on entry BC contains the program
;start address. A test to determine whether the
;computer is a Model I or III(4) is included.
;
$INIT:  PUSH BC         ;start of routine
        XOR A           ;zero A and F
        LD ($EC),A
        LD ($EC+1),A
        LD HL,RETADR    ;an address of a C9.
        LD ($CLSFL),HL
```

```
;
;        Test for MOD I or MOD III
;        store the HIMEM location in MEMLOC
;
         LD HL,011DH      ;ROM addr which contains either
         LD A,76          ;an "L" for Level II or and "M"
         CP (HL)          ;for Model III.
         JR Z,HMI
;
HMIII:   LD HL,(4411H)    ;MOD III HIMEM location
         LD A,62          ;">" = blinker character
         LD B,00H        ;MOD III delay counter(0=256)
         JR SPL
;
HMI:     LD HL,(4049H)    ;MOD I HIMEM location
         LD A,94          ;"->" = blinker character
         LD B,20H         ;MOD I delay counter
;
SPL:
;
         LD DE,CHRPTR     ;store the proper blinker character
         LD (DE),A        ;where it can be retrieved.
         LD DE,FRATE      ;get the flash rate location
         LD A,B           ;transfer rate to A
         LD (DE),A        ;store it
;
         POP BC           ;restore it before we move the stack
                          ;pointer
;
         LD SP,HL         ;reset  stack pointer
         XOR A            ;clear carry flag
                          ;HL still carries the value in HIMEM
         LD DE,1080H      ;4K for displays, 128 bytes for stack
         SBC HL,DE
         LD (DSTART),HL
;
                          ;BC holds the continuation address
         LD H,B           ;put BC into HL
         LD L,C
         JP (HL)          ;start of program
;
RETADR:  RET
```

```
.PRINTX-MENU-
;
; This routine replaces the FORTRAN MENU subroutine often used
; in menu-driven programs.  On entry, HL points to the variable
; ELMNTS, DE points to COLUMN, and BC points to POINTR, all
; integer variables.
;
        VADR EQU 3CBFH          ;start of video memory for Menu
;
;
;
MENU:
        LD (ELMNTS),HL          ;save variables in order:
        LD (COLUMN),DE          ;ELMNTS, COLUMN, POINTR
        LD (POINTR),BC
;
CYCLE1: LD HL,(POINTR)
        LD A,(HL)               ;lower byte of value
        INC HL                  ;next byte
        LD H,(HL)
        LD L,A                  ;now HL = input value in POINTR
        DEC HL                  ;subtract 1
        LD B,6                  ;64 = 2**6
;
CYCLE2: ADD HL,HL               ;double it 6 times until
        DJNZ CYCLE2             ; *64 has been done
;
        LD DE,VADR              ;video offset
        ADD HL,DE               ;position in proper row
        EX DE,HL                ;store it awhile
        LD HL,(COLUMN)
        LD A,(HL)               ;lower byte of value
        INC HL                  ;next byte
        LD H,(HL)
        LD L,A                  ;now HL = input value in COLUMN
        ADD HL,DE               ;current cursor position
        LD (CURPOS),HL          ;store it till later
;
        LD A,(CHRPTR)
        LD (HL),45              ;"-"
        INC HL
        LD (HL),A               ;"->" for Mod I or III
;
        CALL BLINKR             ;the MACRO routine
;
                                ;A contains the key pressed.
        CP 13                   ;look for <Enter>
        RET Z
;
        PUSH AF                 ;save A register
        LD A,(FRATE)            ;start a delay routine
        LD B,A
```

```
SLOW:   LD HL,(CURPOS)          ;need to blank out the arrow
        LD (HL),128             ;with a space
        INC HL
        LD (HL),128
        DJNZ SLOW
;
        POP AF                  ;get the char from BLINKR
        CP 10                   ;"Dn Arrow"
        JR Z,DNTST
;
        CP 91                   ;"Up Arrow"
        JR Z,UPTST
;
        JR CYCLE1               ;start process over
;
DNTST:
        LD DE,(POINTR)
        LD A,(DE)
        LD HL,(ELMNTS)          ;point HL to ELMNTS
        CP (HL)
        JP P,CYCLE1             ;no change if POINTR>=ELMNTS
        EX DE,HL                ;get POINTR
        LD A,1
        ADD A,(HL)
        LD (HL),A               ;POINTR = POINTR + 1
        JR CYCLE1
;
UPTST:
        LD DE,(POINTR)
        LD A,(DE)
        CP 2
        JP M,CYCLE1             ;no change if POINTR=1
        EX DE,HL                ;get POINTR
        LD A,0FFH               ; -1
        ADD A,(HL)
        LD (HL),A               ;POINTR = POINTR - 1
        JR CYCLE1
```

```
.PRINTX-BLINKR-
;
; This routine replaces the FORTRAN BLINKR subroutine often used
; in menu-driven programs.  Because  it is only called from MENU,
; both MENU and BLINKR are more closely coupled, than under
; the FORTRAN calling conventions.
;
;
BLINKR:
;
        LD HL,(CURPOS)          ;current cursor position
        INC HL                  ;next byte = arrowhead
        LD (HL),128
;
NEXT1:  CALL CLBUFF             ;clear the buffer
;
        LD B,80H                ;delay count to flash slowly
;
CYCLE3: CALL 002BH              ;scan once each time for
        OR A                    ;key depression
        RET NZ                  ;go check it out for value
        DJNZ CYCLE3
;
        BIT 7,(HL)              ;test for 128 or ASCII
;
        JR Z,BLINKR             ;go to the 128 unit
;
        LD A,(CHRPTR)           ;get arrow char.
        LD (HL),A
        JR NEXT1                ;go to the next test
```

```
.PRINTX-FCHECK-
;
; FCHECK can be called from a FORTRAN program to test for the
; existence of a data file.  The syntax is:
:       CALL FCHECK(EXIST,FILN)
; where EXIST is a LOGICAL variable which is returned by
; FCHECK and FILN is a LOGICAL array which contains the name
; of the file whose existence is uncertain.  If the file is
; found then FCHECK returns EXIST=.TRUE., otherwise
; EXIST = .FALSE.
;
OPEN    EQU     4424H
FSPEC   EQU     441CH
CLOSE   EQU     4428H
;
FCHECK: PUSH    BC                      ;save all registers
        PUSH    DE
        PUSH    AF
        PUSH    HL
```

```
;
        EX      DE,HL           ;move string pointer to HL
        LD      DE,FCB          ;point to beginning of FCB
        CALL    FSPEC           ;move filespec to FCB
        JR      NZ,ERROUT       ;check for error in filespec
        LD      HL,0            ;A NONSENSE BUFFER
        CALL    OPEN            ;OPEN EXISTING FILE
        PUSH    AF              ;SAVE FLAG
        CALL    CLOSE           ;CLOSE FILE
        POP     AF              ;RECALL FLAG
;
ERROUT: POP     HL              ;RECALL PARAMETER ADDRESS
        JR      Z,OK            ;SKIP IF NO ERROR
        LD      A,0             ;FORTRAN -FALSE-
        LD      (HL),A          ;STORE IT IN PARAMETER
        JR      RETURN          ;GOTO TO RETURN
;
OK:     LD      A,0FFH          ;FORTRAN -TRUE-
        LD      (HL),A          ;STORE IT IN PARAMETER
;
RETURN: POP     AF
        POP     DE
        POP     BC
        RET
;



.PRINTX -DOSCLL-
;
;This is the DOS-CALL routine from FORTRAN.  On entry, HL
;points to a buffer, terminated with a 0D.
;
DOSCLL: JP 4419H
;
;
        END
```

```
      BLOCK DATA ESPDAT

C THIS PROGRAM SETS UP DEFAULT VALUES FOR THE ESP DYNAMIC MODEL

       REAL MRAP,MMD
       INTEGER PSEL
       LOGICAL LRAP,B3,B4,B5,B6,B,A

       COMMON /EPPAR/NSECT,DL,PMMD,TEMP,SN,SCA
       COMMON /SECTL/SSCA(10),EFLD(10),RR(10)
       COMMON /RTME/TIME,TAVG,RAPT(10),THD(10),DLDEV, STCKDM,NLNS
       COMMON /RAPS/TIMRAP,THDRAP,CRTTH,D1,D2
       COMMON /RAPI/IRPFLG,IRAP(10),MRAP(10)
       COMMON /DIS/NDIS,NFS,IADR,NPT,OUTDL,OUTBAR,OP,OPBAR
       COMMON /MET/MMD(1),TDK,SCAM(10),E(10),EPSVIS,OMSN
       COMMON /PRT/IPRT,LRAP(10),ISEL,PSEL
       COMMON /SCPRT/B3,B4,B5,B6,B(121),A(9)

       DATA NSECT,DL,PMMD,TEMP,SN,SCA/4,2.0,20.0,300.,
     * 0.1,200.0/
       DATA SSCA,EFLD,RR/10*50.0,10*2.5,10*0.1/
       DATA TIME,TAVG,RAPT,THD,DLDEV,STCKDM,NLNS/0.0,60.,6.,10.,
     * 15.,30.,60.,5*60.,10*0.0,0.1,20.,10/

       DATA TIMRAP,THDRAP,CRTTH,D1,D2/'ACTV','NACT',2.0,'ACTV',
     * 'NACT'/

       DATA MRAP,LRAP,NDIS,NFS,ISEL,PSEL/10*0.0,10*0,1,1,1,2  /

       DATA B3,B4,B5,B6,B,A/X'1B',X'0A',
     * X'14',X'1E',121*'+',9*'*'/

       END
```

CHAPTER 4

GCA/EPA BAGHOUSE MODEL

Adapted by: A. S. Viner

REVISION 1.2

ABSTRACT

This program is an adaptation of the GCA/EPA baghouse model for the TRS-DOS microcomputer. It predicts the average and maximum tubesheet pressure drop and mass penetration based on user-supplied values of dust load, air flow rate, face velocity, baghouse cycle, and dust and fabric properties. The program is divided into two portions: a data entry program for entering required model parameters and the modeling program that reads data files written by the data entry program and performs the modeling calculations.

USER BENEFITS

- The microcomputer implementation of the GCA/EPA baghouse model provides the advantage of easy accessibility. A user can set up and solve a filtration problem with a minimum of effort.
- The results can be plotted on any standard dot matrix printer if desired.
- The data entry program could easily be modified to serve as a front end program for the mainframe version of the model.

HARDWARE SOFTWARE REQUIREMENTS

The baghouse model is written for the TRS-80 Model I,III, or 4 microcomputer running TRSDOS 1.3, NEWDOS/80, or any other compatible operating system. The data entry program has been designed specifically for the TRS-80 microcomputers and cannot be adapted easily to other machines, but the modeling program can be converted for use on other machines with a minimum of effort. Requirements are 48K of RAM, two disk drives, and a printer.

METHOD

The complete baghouse model and algorithm are described by Dennis et al. (1,2). A brief synopsis follows.

The pressure drop across a filter is assumed to behave according to the relation

$$\Delta P = S_e V + K_2 V W \tag{4-1}$$

where

S_e = the effective residual drag

V = the face velocity

K_2 = the specific resistance coefficient for the dust/fabric combination

W = the dust mass per unit area on the filter.

It is convenient to work in terms of drag, so as to remove the effect of filter face velocity. The drag across the filter is

$$S = S_e + K_2 W \ . \tag{4-2}$$

Equation (4-1) is based on the assumption that the dust mass per unit area (W) is constant over the entire surface of the filter. Dennis et al. (1,2) have found that this is not a good assumption; the dust on a filter surface is distributed very nonuniformly. As a result, they designed their model to allow for up to 10 zones on a bag, with each zone having a different dust load. During a cleaning cycle, the zone with the heaviest loading is cleaned completely, leaving the filter bare in that spot. Equation (4-2) is assumed to apply for each individual area (i):

$$S_i = S_e + K_2 W_i \ . \tag{4-3}$$

The total drag across the filter is then calculated as the sum of parallel resistances:

$$S = A(\Sigma\, A_i/S_i)^{-1} \tag{4-4}$$

where

A = the total area of the filter

A_i = the area whose dust load is W_i.

For a multicompartment baghouse, the drag across each compartment is computed then the compartment drags are added as a set of parallel resistances to yield the complete baghouse drag. The overall pressure drop is the product of the baghouse drag and the face velocity.

The prediction of penetration is based on an empirical correlation developed by GCA from laboratory observations. The correlation consists of three terms: a penetration characteristic of a just cleaned bag, a limiting penetration characteristic of a heavily loaded bag, and a background penetration arising from penetration through pinholes. The penetration is at a maximum just after cleaning and decreases exponentially with increasing dust load. The correlation incorporates a velocity dependence so that the penetration increases with increasing face velocity.

GENERAL RESTRICTIONS

Reasonable default values are supplied. Negative values should not be used for any input quantity, and zero values should not be used for quantities representing physical dimensions. The results are unpredictable in those cases.

Measured values of K_2 will give more accurate results than predicted values.

For the printout to appear as in the sample figures, the printer must be set up to print a 132-column line.

SAMPLE PROBLEM

Statement: Determine the average and maximum pressure drop and mass emissions for a baghouse with two compartments and a face velocity of .305 m/min. The compartments are taken off-line and cleaned for 5 minutes before being returned to service. The cleaning cycle time is 10 minutes and the time between cleaning cycles is 5 minutes. The dust load is estimated to be 3.56 g/m^3. The specific resistance coefficient (K_2) is not known but can be estimated from the following fly ash properties:

Mass median diameter	4.0 μm
Standard deviation	2.5 μm
Particle density	2.2 g/cm^3
Bulk density	1.1 g/cm^3 .

The operating temperature of the baghouse is 185 °C. The effective residual drag (S_e) is 350 N-min/m^3 and the residual loading (W_r) is 50 g/m^2.

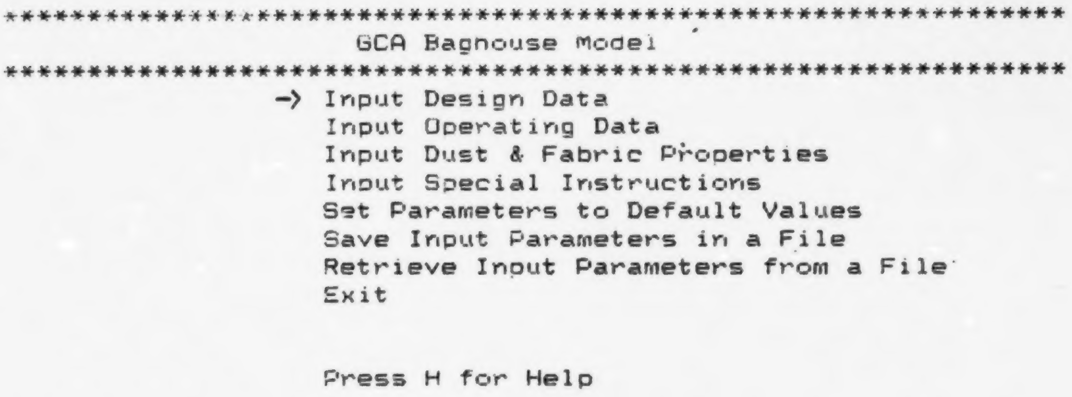
Results: This sample problem will serve as the basis for a tutorial on how to use the baghouse programs. The reader is encouraged to sit in front of the computer and try each step as it is described. It is also recommended that the user compare the results obtained from the program with the results shown below to verify that the program is operating properly.

There are basically two steps involved in the use of this microcomputer model: creating a data file and running the model for that input. The first step is to create a data file containing all the design and operating parameters using the DATIN program. When the DATIN program is run, the menu shown in Figure 4-1 will appear on the screen. Note that pressing <H> will switch the program to the help sequence (Figure 4-2). The first four subroutines listed in the menu are for data input. These subroutines follow the natural breakdown used by Dennis et al. in the original model.

To begin the data entry, call the "Input Design Data" subroutine. The menu shown in Figure 4-3 should appear on the screen. The message at the top of the display identifies the subroutine and indicates that this is the only page of this menu. The parameters and default values are the same as those recommended by Dennis et al. (1,2). Thus, the parameters listed in this design menu are all zero. Also note the "Help" option. Pressing H will yield the page shown in Figure 4-4.

The first row in the menu, labeled "Title," allows the user to enter an identifying message up to 40 characters in length to describe this run. The second row, "Number of Compartments," allows the varying of the number of compartments, or modules, from 2 to 30. The compartment cleaning time is the length of time a compartment remains off-line while it is being cleaned. The cleaning cycle time is the length of time that elapses between the beginning of cleaning of the first compartment and the end of cleaning of the last cleaned compartment. The cleaning cycle time must be greater than the compartment cleaning time.

The last five items in the Design Data menu are used to specify the method and onset of cleaning. Cleaning can be initialized according to a set schedule, or it can be started when a user-specified pressure drop is exceeded. The user can choose either method by specifying a value for either "Time Between Cleaning Cycles" or "Limiting Pressure Drop." The method of cleaning can be either reverse-air or shake deflate. The choice



```
*****************************************************************
                    GCA Bagnouse Model
*****************************************************************
                -> Input Design Data
                   Input Operating Data
                   Input Dust & Fabric Properties
                   Input Special Instructions
                   Set Parameters to Default Values
                   Save Input Parameters in a File
                   Retrieve Input Parameters from a File
                   Exit


                   Press H for Help
```

Figure 4-1. The Main Menu of the DATIN program.

THE PREVIOUS PAGE LISTED 8 COMMANDS AVAILABLE TO YOU: 4 DATA INPUTS; SET PARAMETERS TO DEFAULT VALUES, STORE/ RETRIEVE DATA TO/FROM A DISK FILE; AND BEGIN MODELLING. TO EXECUTE ANY OF THESE COMMANDS MOVE THE POINTER (→) TO THE PROPER ROW USING THE ↑ AND ↓ KEYS AND SELECT THE COMMAND BY PRESSING THE ⟨ENTER⟩ KEY. FOR EXAMPLE TO ENTER DESIGN DATA MOVE THE POINTER TO THE ROW LABELLED "INPUT DESIGN DATA" AND PRESS THE ⟨ENTER⟩ KEY. THE DESIGN DATA WILL BE DISPLAYED ON THE SCREEN AND INSTRUCTIONS WILL BE GIVEN ON HOW TO ENTER DATA.

PRESS ANY KEY TO CONTINUE

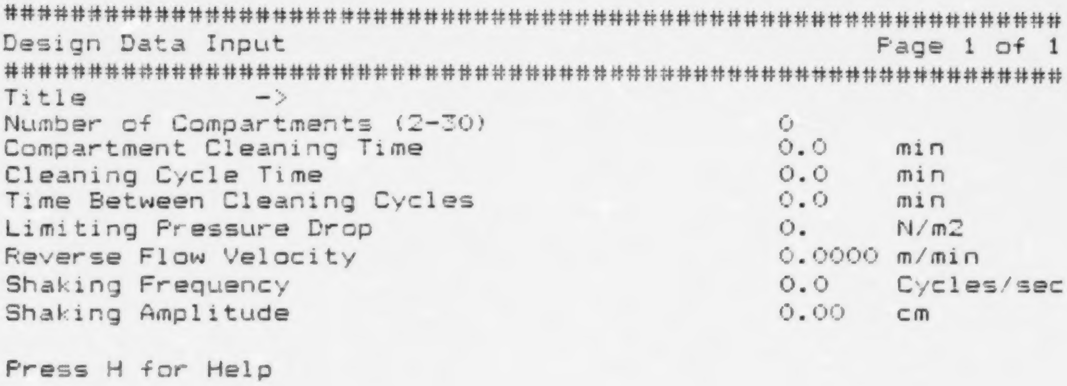
WHEN THE PROGRAM IS FIRST STARTED, THE PARAMETERS ARE INITIALIZED WITH THE VALUES GIVEN IN EXAMPLE 3 ON PAGE 160 OF THE DETAILED TECHNICAL REPORT. IF AFTER RUNNING SEVERAL TEST CASES YOU WISH TO RETURN TO THESE DEFAULT VALUES THEN USE THE "SET . . . DEFAULT VALUES" SUBROUTINE.

ONCE THE DATA HAS BEEN ENTERED IT CAN THEN BE SAVED ONTO A DISK TO BE RECALLED AT A LATER TIME. THESE COMMANDS ARE EXECUTED IN THE SAME WAY; I.E. POSITION THE POINTER AND PRESS ⟨ENTER⟩.

WHEN YOU ARE READY TO EXECUTE THE PROGRAM MOVE THE POINTER TO THE ROW LABELLED "BEGIN PERFORMANCE MODELLING" AND SELECT THIS ROUTINE. RESULTS OF THE MODELLING WILL BE DISPLAYED ON THE SCREEN AND YOU WILL HAVE THE OPTION TO PRINT OUT THE RESULTS ON THE PRINTER.

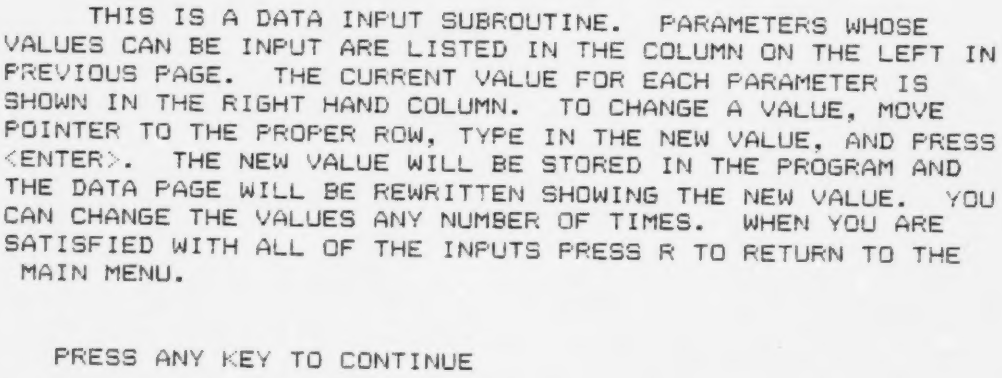
PRESS ANY KEY TO CONTINUE

Figure 4-2. "Help" pages called from Main Menu of DATIN program.



```
##############################################################
Design Data Input                                   Page 1 of 1
##############################################################
Title          ->
Number of Compartments (2-30)                 0
Compartment Cleaning Time                     0.0     min
Cleaning Cycle Time                           0.0     min
Time Between Cleaning Cycles                  0.0     min
Limiting Pressure Drop                        0.      N/m2
Reverse Flow Velocity                         0.0000 m/min
Shaking Frequency                             0.0     Cycles/sec
Shaking Amplitude                             0.00    cm

Press H for Help
```

Figure 4-3. "Input Design Data" menu.



```
     THIS IS A DATA INPUT SUBROUTINE.  PARAMETERS WHOSE
VALUES CAN BE INPUT ARE LISTED IN THE COLUMN ON THE LEFT IN
PREVIOUS PAGE.  THE CURRENT VALUE FOR EACH PARAMETER IS
SHOWN IN THE RIGHT HAND COLUMN.  TO CHANGE A VALUE, MOVE
POINTER TO THE PROPER ROW, TYPE IN THE NEW VALUE, AND PRESS
<ENTER>.  THE NEW VALUE WILL BE STORED IN THE PROGRAM AND
THE DATA PAGE WILL BE REWRITTEN SHOWING THE NEW VALUE.  YOU
CAN CHANGE THE VALUES ANY NUMBER OF TIMES.  WHEN YOU ARE
SATISFIED WITH ALL OF THE INPUTS PRESS R TO RETURN TO THE
 MAIN MENU.

   PRESS ANY KEY TO CONTINUE
```

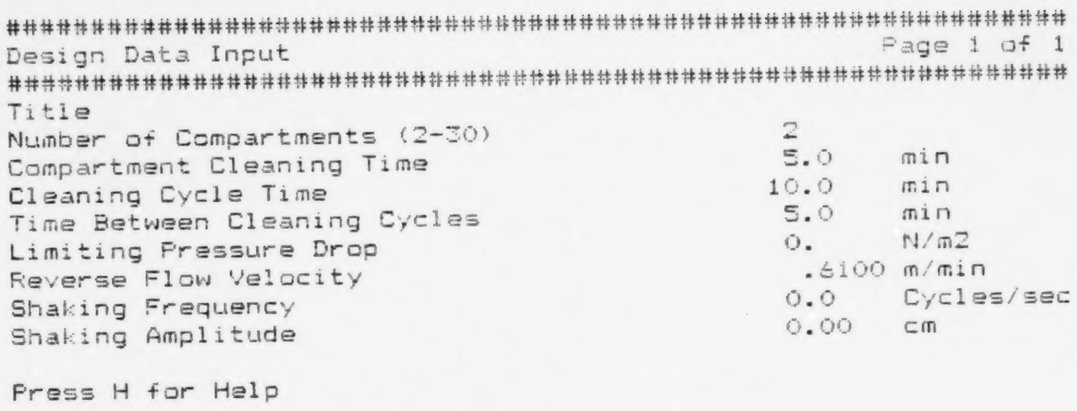
Figure 4-4. Help routine called from "Design Data" menu.

is made by assigning a nonzero value for "Reverse Flow Velocity" or for "Shaking Frequency" and "Shaking Amplitude."

The sample problem calls for a baghouse with two compartments and a compartment cleaning time of 5 minutes. In addition, the baghouse will be cleaned by the reverse air method, with 5 minutes between the 10-minute cleaning cycles and a reverse air velocity of 0.61 m/min. Begin entering these values by positioning the pointer to the right of the "Number of Compartments" using the up and down arrows. Then type 2 and press <ENTER>. This will change the value from 0 to 2. Next enter the other values (the screen should resemble Figure 4-5) and return to the Main Menu by pressing <R>.

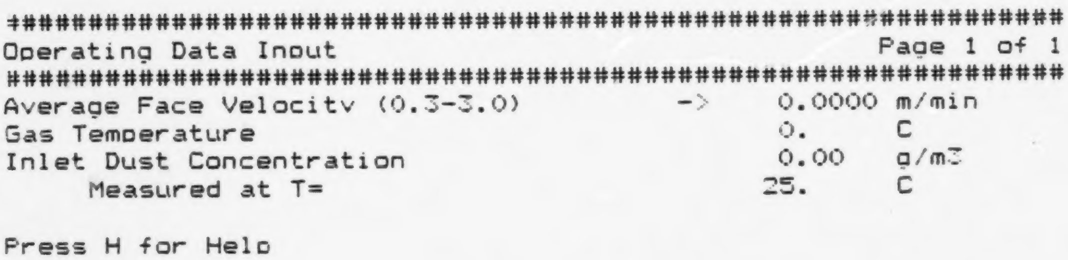
The second part of the data entry process is to enter the operating data. This is done by calling the "Input Operating Data" option from the Main Menu. The menu shown in Figure 4-6 will appear on the screen. The items in this menu describe the conditions as seen by the filter bags. The face velocity is the ratio of the flue gas flow rate to the net fabric area. The inlet dust concentration should be the measured dust load at the tubesheet. For this sample problem the face velocity is .305 m/min at a gas temperature of 185 °C. The inlet dust concentration is 3.56 g/m^3 measured at 185 °C. When these values have been correctly entered into the menu, press <R> to return to the Main Menu.

The third group of baghouse input data are entered through the "Input Dust and Fabric Properties" option of the Main Menu. When the "Input Dust and Fabric Properties" option is selected, the screen should appear as in Figure 4-7. This menu is different from the previous menus because it has two pages. The procedure for entering a value remains the same. In addition, the program allows you to go back and forth between the pages by using the < or > keys. Figure 4-7 is the first page of this menu and contains the data pertaining to the Specific Resistance coefficient (K_2). If a value of K_2 has been measured, it can be entered as the first item in the menu. The next two items below the K_2 entry are for the temperature and the face velocity of the gas during the measurement of the K_2 value. Also, if the K_2 is measured with a dust that is different from the dust to be modelled, the mass mean diameter (MMD) and geometric standard deviation (std. dev.) of the test dust as well as that of the inlet dust (the last two items on page 1)



```
############################################################
Design Data Input                                Page 1 of 1
############################################################
Title
Number of Compartments (2-30)                     2
Compartment Cleaning Time                         5.0    min
Cleaning Cycle Time                              10.0    min
Time Between Cleaning Cycles                      5.0    min
Limiting Pressure Drop                            0.     N/m2
Reverse Flow Velocity                              .6100 m/min
Shaking Frequency                                 0.0    Cycles/sec
Shaking Amplitude                                 0.00   cm

Press H for Help
```

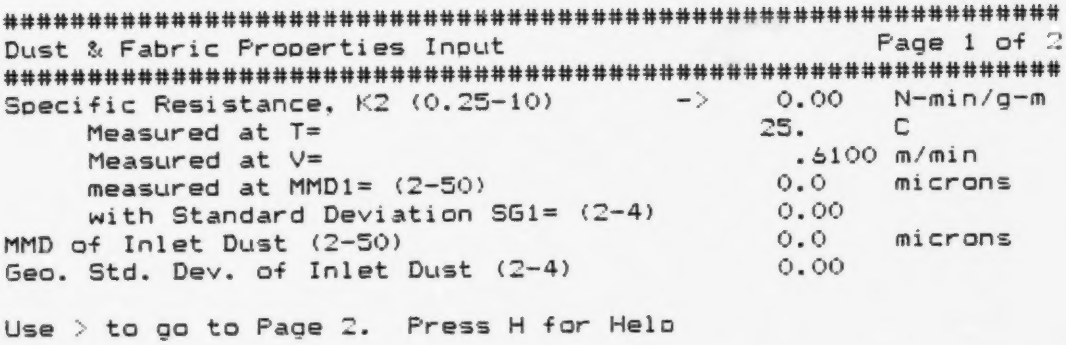
Figure 4-5. "Design Data" menu after data entry.



```
############################################################
Ooerating Data Inout                             Page 1 of 1
############################################################
Average Face Velocitv (0.3-3.0)        ->     0.0000 m/min
Gas Temoerature                                0.     C
Inlet Dust Concentration                       0.00   g/m3
     Measured at T=                           25.     C

Press H for Helo
```

Figure 4-6. "Input Operating Data" menu.

146



```
##################################################################
Dust & Fabric Properties Input                       Page 1 of 2
##################################################################
Specific Resistance, K2 (0.25-10)      ->     0.00    N-min/g-m
     Measured at T=                          25.      C
     Measured at V=                            .6100 m/min
     measured at MMD1= (2-50)                 0.0     microns
     with Standard Deviation SG1= (2-4)       0.00
MMD of Inlet Dust (2-50)                      0.0     microns
Geo. Std. Dev. of Inlet Dust (2-4)            0.00

Use > to go to Page 2.  Press H for Help
```

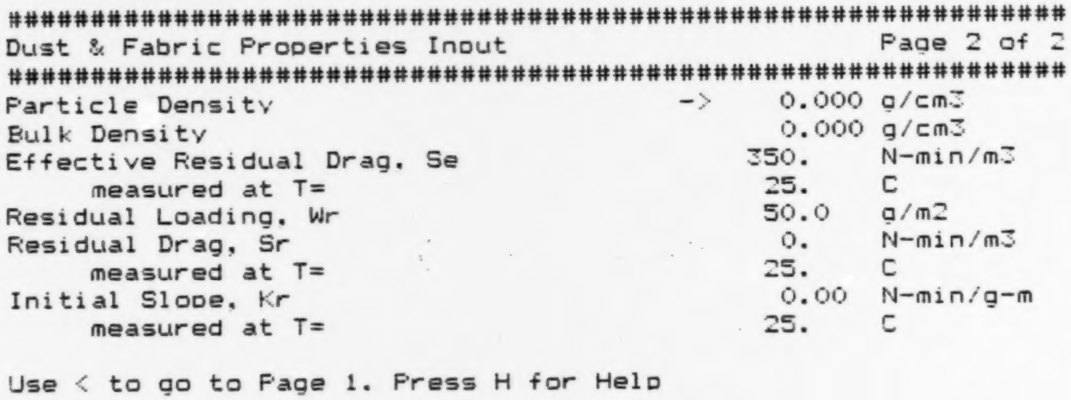
Figure 4-7. Page 1 of "Input Dust & Fabric Properties" menu.

should be specified. These values must be in the range of 2 to 50 μm and 2 to 4 μm, respectively. The baghouse program will adjust the measured value of K_2 to account for differences in specific surface area of the test dust and the inlet dust.

For this sample problem, enter the MMD of 4.0 μm and a std. dev. of 2.5, then go on to page 2. The menu shown in Figure 4-8 will appear on the screen. The first two parameters are used in the calculation of K_2 if a value was not specified on the previous page (as in this problem). The "Bulk Density" refers to the density of the dust cake. The remaining parameters in this display are used to describe the drag across the dust/fabric combination just after cleaning. This is accomplished using either of the two models, linear or nonlinear, that are available to describe the pressure rise after cleaning. To specify the linear model, the user must enter values for S_e, the effective residual drag, the temperature at which S_e was measured, and the residual dust load, W_r. The nonlinear model will be used if the user also enters values for the measured residual drag (S_r), the initial resistance coefficient (K_r), and the temperature at which S_r and K_r were measured. For this sample problem, enter the particle density as 2.2 g/cm^3, the bulk density as 1.1 g/cm^3, S_e as 350 N-min/m^3, and the temperature as 185 °C (see problem statement).

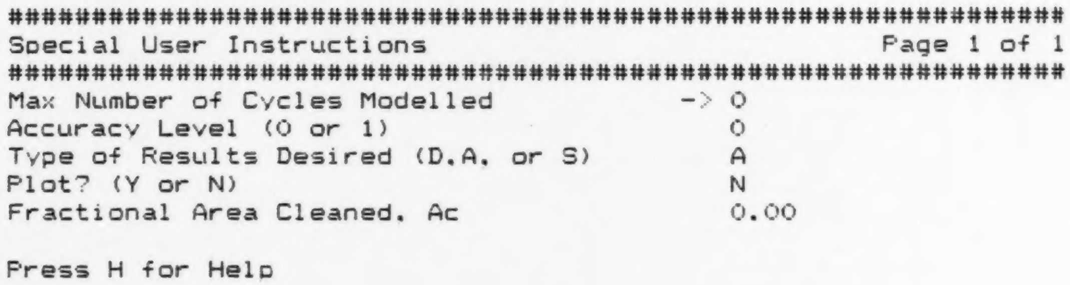
The last input routine listed in the Main Menu is called "Input Special Instructions." When this option is called, the menu shown in Figure 4-9 will appear on the screen. These parameters give the user control of some aspects of the program. For example, the "Max Number of Cycles Modelled" is, in essence, a time limit for how long the program can run. If the modeling program does not reach steady state within three cycles of the specified number, then the results consist of a warning message along with the performance characteristics of the last three cycles. If steady state is reached, the program prints out the results of the first three steady-state cycles. Enter a value of 20. The "Accuracy Level" shown in the program allows the user to choose between the levels of 1 and 0 (1 is the more stringent choice). Choose 0.

"Type of Results Desired" allows the user to choose between three levels of detail for the output of the program: Detailed(D), Summary(S), or Average(A). The default value is A. For this sample problem choose D. The



```
################################################################
Dust & Fabric Properties Input                        Page 2 of 2
################################################################
Particle Density                     ->     0.000 g/cm3
Bulk Density                                0.000 g/cm3
Effective Residual Drag, Se                350.   N-min/m3
     measured at T=                         25.   C
Residual Loading, Wr                        50.0  g/m2
Residual Drag, Sr                            0.   N-min/m3
     measured at T=                         25.   C
Initial Slope, Kr                            0.00 N-min/g-m
     measured at T=                         25.   C

Use < to go to Page 1. Press H for Help
```

Figure 4-8. Page 2 of "Input Dust & Fabric Properties" menu.



```
################################################################
Special User Instructions                             Page 1 of 1
################################################################
Max Number of Cycles Modelled           -> 0
Accuracy Level (0 or 1)                    0
Type of Results Desired (D,A, or S)        A
Plot? (Y or N)                             N
Fractional Area Cleaned, Ac                0.00

Press H for Help
```

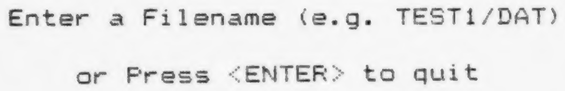
Figure 4-9. "Input Special Instructions" menu.

next option, "Plot," allows the user to choose whether to have the data plotted. Designate a choice by typing in <Y>. "Fractional Area Cleaned" can be used to specify the amount of the bag surface that is completely cleaned during each cleaning cycle. Leave this parameter at its default value.

When all the values have been entered, return to the Main Menu then use the "Save Input Parameters in a File" option to save these data. When this option is chosen, the program will ask for the name of a file in which to store the data. See Figure 4-10. Entering a name allows the program to save the data. In this sample problem the name TEST1/DAT is used. If at a later time the user wishes to change one or more of the values for another run, the "Retrieve Input Parameters from a File" option of the Main Menu allows the retrieval of data previously stored. This can then be modified and stored under the same name or as a new file. A hard copy option is also available. Figure 4-11 is the printout for this problem. Obtain a hard copy and compare it to Figure 4-11. If differences are found, return to the various data menus and correct the values, then save the file again. The last option, "Set Parameters to Default Values," can be used to return the values of the parameters to their default values.

When the input data file(s) have been created with the DATIN Program, exit the program; then the user can run the modelling program. There are two steps required before the program can be run. First, attach and turn on a printer. Second, the user must protect some of the computer's memory locations. If the TRSDOS 1.3 operating system is being used, the command is CLEAR (MEM=OF7FF) from DOS READY. The procedure for protecting memory varies from one operating system to another; please consult your documentation.

When the printer is ready and the memory locations have been protected, the baghouse modelling program can be run. The only input required of the user is the name of the file in which the input parameters are stored. After the file name has been entered the program reads the data file, does some preliminary calculations, and prints some calculated values on the printer (see Figure 4-12 for printer output). Then the program will begin iterating until the steady-state pressure drop cycle has been found. This example will take almost 8 minutes. The computer screen displays the results as a simulated strip chart recorder. That is, the system pressure drop is scaled and plotted as a dot in the rightmost column of the screen.

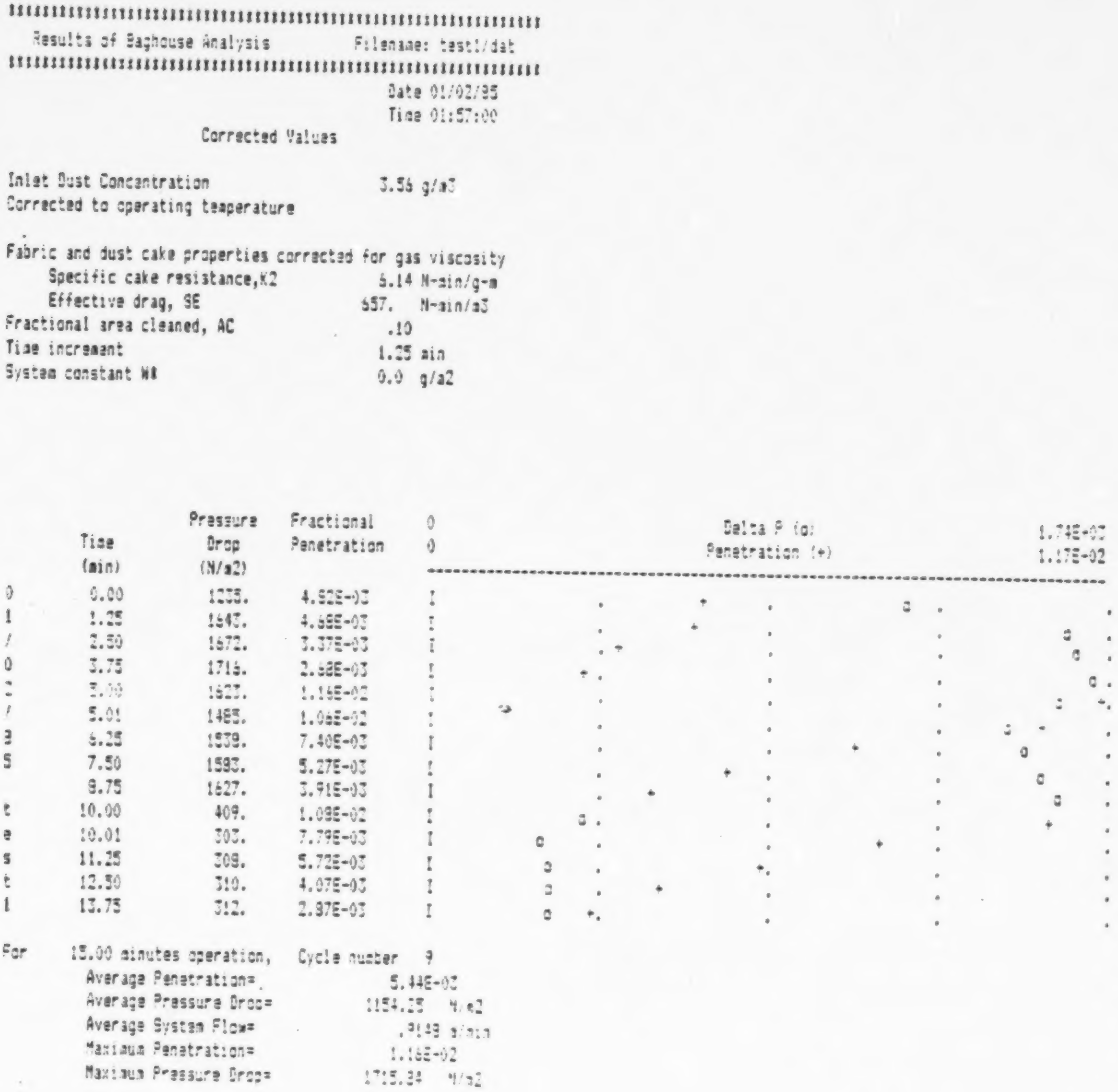


```
Enter a Filename (e.g. TEST1/DAT)

    or Press <ENTER> to quit
```

Figure 4-10. Request for file name.

```
################################################################################

Summary of Input Data For   Baghouse Analysis        Filename: TEST1/DAT

################################################################################

                                                          Date 02/03/85
                                                          Time 00:06:11


Basic Design Data
     Number of Compartments                                 2
     Compartment Cleaning Time                              5.0    min
       (Off Line Time)
     Cleaning Cycle Time                                   10.0    min
     Time Between Cleaning Cycles                           5.0    min
     Reverse Flow Velocity                                   .6100 m/min


Operating Data
     Average Face Velocity                                   .3050 m/min
     Gas Temperature                                      185.     C
     Inlet Dust Concentration                               3.56   g/m3
          measured at T=                                  185.     C


Fabric and Dust Properties
     Specific Resistance, K2 Estimated from
          Mass Median Diameter                              4.0    microns
          Standard Deviation                                2.50
          Particle Density                                  2.200  g/cm3
          Bulk Density                                      1.100  g/cm3
     Effective Residual Drag, Se                          350.     N-min/m3
          measured at T=                                  185.     C
     Residual Loading, Wr                                  50.0    g/m2


Special Program Instructions
     Max Number of Cycles Modelled                          20
     Accuracy Level                                          0
     Type of Results Desired                                 D
     Plot ?                                                  Y
```

Figure 4-11. Printout from DATIN for sample problem.



```
################################################################
   Results of Baghouse Analysis        Filename: test1/dat
################################################################
                                          Date 01/02/85
                                          Time 01:57:00
                       Corrected Values

Inlet Dust Concentration                  3.56 g/m3
Corrected to operating temperature

Fabric and dust cake properties corrected for gas viscosity
     Specific cake resistance,K2          6.14 N-min/g-m
     Effective drag, SE                  657.   N-min/m3
Fractional area cleaned, AC                .10
Time increment                            1.25 min
System constant W*                        0.0  g/m2



                  Pressure   Fractional   0                                  Delta P (o)                           1.74E+03
        Time      Drop       Penetration  0                                 Penetration (+)                        1.17E-02
       (min)     (N/m2)                   ---------------------------------------------------------------------------------
0        0.00       1233.    4.92E-03     I                   .           +       .               o   .                   .
1        1.25       1643.    4.66E-03     I                   .          +        .                   .              o    .
/        2.50       1672.    3.37E-03     I                   . +                 .                   .               o   .
0        3.75       1716.    2.68E-03     I                 + .                   .                   .                 o .
2        5.00       1623.    1.16E-02     I        +          .                   .                   .             o    +.
/        5.01       1485.    1.06E-02     I                   .                   .                   .       o   +       .
8        6.25       1539.    7.40E-03     I                   .                   .         +         .         o         .
5        7.50       1593.    5.27E-03     I                   .              +    .                   .           o       .
         8.75       1627.    3.91E-03     I                   .     +             .                   .             o     .
t       10.00        409.    1.09E-02     I                 o .                   .                   .            +      .
e       10.01        303.    7.79E-03     I            o      .                   .            +      .                   .
s       11.25        309.    5.72E-03     I             o     .                  +.                   .                   .
t       12.50        310.    4.07E-03     I             o     .      +            .                   .                   .
1       13.75        312.    2.87E-03     I             o    +.                   .                   .                   .

For      15.00 minutes operation,   Cycle number  9
          Average Penetration=                5.44E-03
          Average Pressure Drop=           1154.25  N/m2
          Average System Flow=               .9149 m/min
          Maximum Penetration=                1.16E-02
          Maximum Pressure Drop=           1715.34  N/m2
```

Figure 4-12. Results for sample problem.

Reproduced from best available copy.

When the next value of the system pressure drop is determined, the contents of the entire display are moved left by one column and the current pressure drop is plotted. The system pressure drop is plotted for every time increment. Since a plot was requested the program produces a simple printer plot of the system pressure drop and penetration for one filtration cycle. Since the program prints the results as they are calculated, the time to print the results is equal to the running time of one baghouse cycle (cleaning + filtration time).

The biggest disadvantage of the GCA/EPA baghouse model is that it can take hours to converge on the steady-state solution. The running time depends directly on the compartment cleaning time, the cleaning cycle length, and the filtration time. If the cleaning cycle time is short in relation to the overall cycle time, the program will take a long time to run. The run time varies from minutes to hours, depending on these values. Research is underway to improve this aspect of the model.

This ends the sample problem. The user may now return to the DATIN program and change values or options to determine the effect on the output.

REFERENCES

1. Dennis, R., et al. Filtration Model for Coal Fly Ash with Glass Fabrics. EPA-600/7-77-084 (NTIS PB 276489). August 1977.

2. Dennis, R., and H. A. Klemm. Fabric Filter Model Format Change: Volumes I and II. EPA-600/7-79-043a and b (NTIS PB 293551 and 294042). February 1979.

```
      PROGRAM DATIN
      DIMENSION LABEL(40),HEAD(40)
      REAL MMD1,MMD2
      INTEGER HEAD,POINTR,ELMNTS,ROW,COLUMN,DATYPE,PLTYPE
      LOGICAL ERROR,ALDIAG,PLDIAG,PRDIAG,HOME,FILN(13),SFLAG(10),JUNK
      LOGICAL HCOPY,NEW
      COMMON /DESIGN/N,T,TCLEAN,TLAG,VRFLO,DPSTOP,FREQ,AMPLIT
      COMMON /TITLE/HEAD
      COMMON /OPDATA/SMALQ,TEMPK,CZERO,TCZERO
      COMMON /FABDUS/ZK2,SE,WR,SR,ZKR,WSTAR
      COMMON /K2EST/TZK2,VZK2,MMD1,SG1,MMD2,SG2,RHOP,RHOBLK
      COMMON /MUCORR/TSE,TSR,TZKR
      COMMON /ACURAC/JCODE
      COMMON /EXTERN/NT,M,WSTART,ACLEAN
      COMMON /OUTPUT/DATYPE,PLTYPE,HCOPY
      COMMON /DIAG/ALDIAG,PRDIAG,PLDIAG
      DATA HEAD,ALDIAG,PLDIAG,PRDIAG/40*32,3*.FALSE./
      CALL DFAULT
      JPOINT=1
      NEW=.FALSE.
      HOME=X'1C'
1     CALL CLS
      DO 4 I=1,13
      FILN(I)=32
      IF(I.GT.10)GOTO 4
      SFLAG(I)=.TRUE.
4     CONTINUE
      WRITE(3,100)HOME
100   FORMAT(
     1 '+',1A1,63('*')
     2 /'+',22X,'GCA Baghouse Model'
     3 /'+',63('*')
     4 /'+',20X,'Input Design Data'
     5 /'+',20X,'Input Operating Data'
     6 /'+',20X,'Input Dust & Fabric Properties'
     7 /'+',20X,'Input Special Instructions'
     8 /'+',20X,'Set Parameters to Default Values'
     9 /'+',20X,'Save Input Parameters in a File'
     A /'+',20X,'Retrieve Input Parameters from a File'
     B /'+',20X,'Exit'
     D //
     E /'+',20X,'Press H for Help')
      POINTR=JPOINT
      ELMNTS=8
2     CALL MENU(ELMNTS,POINTR,SFLAG,ROW,COLUMN,LETTER)
      IF(LETTER.EQ.72)CALL HELP
      IF(LETTER.EQ.72)GOTO 1
      IF(LETTER.NE.13)GOTO 2
      DO 3 I=1,10
3     SFLAG(I)=.FALSE.
      JPOINT=POINTR
      GOTO(10,20,30,40,50,60,60,80),POINTR
10    POINTR=1
      ELMNTS=9
      SFLAG(1)=.TRUE.
```

```
11      CALL CLS
        WRITE(3,110)HOME
110     FORMAT('+',1A1,63('#')
      2 /'+Design Data Input',35X,'Page 1 of 1'
      3 /'+',63('#'))
        WRITE(3,111)N,TCLEAN,T,TLAG,DPSTOP,VRFLO,FREQ,AMPLIT
111     FORMAT(
      5 '+Title'
      6 /'+Number of Compartments (2-30)',16X,I2
      7 /'+Compartment Cleaning Time',18X,F6.1,'    min'
      8 /'+Cleaning Cycle Time',24X,F6.1,'    min'
      8 /'+Time Between Cleaning Cycles',15X,F6.1,'    min'
      9 /'+Limiting Pressure Drop',21X,F5.0,'     N/m2'
      A /'+Reverse Flow Velocity',24X,F7.4,' m/min'
      B /'+Shaking Frequency',28X,F4.1,'    Cycles/sec'
      C /'+Shaking Amplitude',28X,F5.2,'   cm'
      D /
      E /'+Press H for Help')
        DO 13 I=1,40
        K=I+15568
13      CALL POKE(K,HEAD(I))
12      CALL MENU(ELMNTS,POINTR,SFLAG,ROW,COLUMN,LETTER)
        IF(LETTER.EQ.82)GOTO 1
        IF(LETTER.EQ.72)CALL HELP1
        IF(LETTER.EQ.72)GOTO 11
        IF(.NOT.(((LETTER.GE.48).AND.(LETTER.LE.57))
      1 .OR.(LETTER.EQ.45)
      2 .OR.(LETTER.EQ.46)))GOTO 14
        Z=VALUE(ROW,COLUMN,LETTER)
        IF(POINTR.EQ.2)N=INT(Z)
        IF(POINTR.EQ.3)TCLEAN=Z
        IF(POINTR.EQ.4)T=Z
        IF(POINTR.EQ.5)TLAG=Z
        IF(POINTR.EQ.6)DPSTOP=Z
        IF(POINTR.EQ.7)VRFLO=Z
        IF(POINTR.EQ.8)FREQ=Z
        IF(POINTR.EQ.9)AMPLIT=Z
        GOTO 11
14      IF(POINTR.NE.1)GOTO 12
        HEAD(1)=LETTER
        COLUMN=17
        CALL STRNG(HEAD,ROW,COLUMN,LETTER,NCHAR)
        GOTO 11
20      POINTR=1
        ELMNTS=4
21      CALL CLS
        WRITE(3,120)HOME
120     FORMAT('+',1A1,63('#')
      2 /'+Operating Data Input',32X,'Page 1 of 1'
      3 /'+',63('#'))
        WRITE(3,121)SMALQ,TEMPK,CZERO,TCZERO
121     FORMAT(
      5 '+Average Face Velocity (0.3-3.0)',14X,F7.4,' m/min'
      6 /'+Gas Temperature',28X,F5.0,'     C'
      7 /'+Inlet Dust Concentration',20X,F6.2,'   g/m3'
```

```
          8 /'+     Measured at T=',24X,F5.0,'     C'
          9 /
          A /'+Press H for Help')
    22      CALL MENU(ELMNTS,POINTR,SFLAG,ROW,COLUMN,LETTER)
            IF(LETTER.EQ.82)GOTO 1
            IF(LETTER.EQ.72)CALL HELP1
            IF(LETTER.EQ.72)GOTO 21
            IF(.NOT.(((LETTER.GE.48).AND.(LETTER.LE.57))
          1 .OR.(LETTER.EQ.45)
          2 .OR.(LETTER.EQ.46)))GOTO 22
            Z=VALUE(ROW,COLUMN,LETTER)
            IF(POINTR.EQ.1)SMALQ=Z
            IF(POINTR.EQ.2)TEMPK=Z
            IF(POINTR.EQ.3)CZERO=Z
            IF(POINTR.EQ.4)TCZERO=Z
            GOTO 21
    30      POINTR=1
            ELMNTS=7
    31      CALL CLS
            WRITE(3,130)HOME
    130     FORMAT('+',1A1,63('#')
          2 /'+Dust & Fabric Properties Input',22X,'Page 1 of 2'
          3 /'+',63('#'))
            WRITE(3,131)ZK2,TZK2,VZK2,MMD1,SG1,MMD2,SG2
    131     FORMAT(
          5 '+Specific Resistance, K2 (0.25-10)',11X,F6.2,'   N-min/g-m'
          6 /'+     Measured at T=',24X,F5.0,'     C'
          7 /'+     Measured at V=',26X,F7.4,' m/min'
          8 /'+     measured at MMD1= (2-50)',16X,F4.1,'    microns'
          9 /'+     with Standard Deviation SG1= (2-4)',7X,F4.2
          A /'+MMD of Inlet Dust (2-50)',21X,F4.1,'    microns'
          B /'+Geo. Std. Dev. of Inlet Dust (2-4)',12X,F4.2
          C /
          D /'+Use ) to go to Page 2.  Press H for Help')
    32      CALL MENU(ELMNTS,POINTR,SFLAG,ROW,COLUMN,LETTER)
            IF(LETTER.EQ.82)GOTO 1
            IF(LETTER.EQ.62)GOTO 33
            IF(LETTER.EQ.72)CALL HELP2
            IF(LETTER.EQ.72)GOTO 31
            IF(.NOT.(((LETTER.GE.48).AND.(LETTER.LE.57))
          1 .OR.(LETTER.EQ.45)
          2 .OR.(LETTER.EQ.46)))GOTO 32
            Z=VALUE(ROW,COLUMN,LETTER)
            IF(POINTR.EQ.1)ZK2=Z
            IF(POINTR.EQ.2)TZK2=Z
            IF(POINTR.EQ.3)VZK2=Z
            IF(POINTR.EQ.4)MMD1=Z
            IF(POINTR.EQ.5)SG1=Z
            IF(POINTR.EQ.6)MMD2=Z
            IF(POINTR.EQ.7)SG2=Z
            GOTO 31
    33      POINTR=1
            ELMNTS=9
    34      CALL CLS
            WRITE(3,135)HOME
```

```
 135    FORMAT('+',1A1,63('#')
     2  /'+Dust & Fabric Properties Input',22X,'Page 2 of 2'
     3  /'+',63('#'))
        WRITE(3,136)RHOP,RHOBLK,SE,TSE,WR,SR,TSR,ZKR,TZKR
 136    FORMAT(
     5  '+Particle Density',29X,F6.3,' g/cm3'
     6  /'+Bulk Density',33X,F6.3,' g/cm3'
     7  /'+Effective Residual Drag, Se',16X,F5.0,'    N-min/m3'
     8  /'+     measured at T=',24X,F5.0,'    C'
     9  /'+Residual Loading, Wr',23X,F6.1,'   g/m2'
     A  /'+Residual Drag, Sr',26X,F5.0,'    N-min/m3'
     B  /'+     measured at T=',24X,F5.0,'    C'
     C  /'+Initial Slope, Kr',27X,F6.2,'  N-min/g-m'
     D  /'+     measured at T=',24X,F5.0,'    C'
     E  /
     F  /'+Use ( to go to Page 1. Press H for Help')
 35     CALL MENU(ELMNTS,POINTR,SFLAG,ROW,COLUMN,LETTER)
        IF(LETTER.EQ.82)GOTO 1
        IF(LETTER.EQ.60)GOTO 30
        IF(LETTER.EQ.72)CALL HELP2
        IF(LETTER.EQ.72)GOTO 34
        IF(.NOT.(((LETTER.GE.48).AND.(LETTER.LE.57))
     1  .OR.(LETTER.EQ.45)
     2  .OR.(LETTER.EQ.46)))GOTO 35
        Z=VALUE(ROW,COLUMN,LETTER)
        IF(POINTR.EQ.1)RHOP=Z
        IF(POINTR.EQ.2)RHOBLK=Z
        IF(POINTR.EQ.3)SE=Z
        IF(POINTR.EQ.4)TSE=Z
        IF(POINTR.EQ.5)WR=Z
        IF(POINTR.EQ.6)SR=Z
        IF(POINTR.EQ.7)TSR=Z
        IF(POINTR.EQ.8)ZKR=Z
        IF(POINTR.EQ.9)TZKR=Z
        GOTO 34
 40     POINTR=1
        ELMNTS=5
 41     CALL CLS
        WRITE(3,140)HOME
 140    FORMAT('+',1A1,63('#')
     2  /'+Special User Instructions',27X,'Page 1 of 1'
     3  /'+',63('#'))
        WRITE(3,141)NT,JCODE,DATYPE,PLTYPE,ACLEAN
 141    FORMAT(
     5  '+Max Number of Cycles Modelled',13X,I2
     6  /'+Accuracy Level (0 or 1)',20X,I1
     7  /'+Type of Results Desired (D,A, or S)',8X,A1
     8  /'+Plot? (Y or N)',29X,A1
     9  /'+Fractional Area Cleaned, Ac',16X,F4.2
     9  /
     9  /'+Press H for Help')
 42     CALL MENU(ELMNTS,POINTR,SFLAG,ROW,COLUMN,LETTER)
        IF(LETTER.EQ.82)GOTO 1
        IF(LETTER.EQ.72)CALL HELP1
        IF(LETTER.EQ.72)GOTO 41
```

```
      GOTO (43,44,45,46,43),POINTR
43    IF(.NOT.(((LETTER.GE.48).AND.(LETTER.LE.57))
     1 .OR.(LETTER.EQ.45)
     2 .OR.(LETTER.EQ.46)))GOTO 42
      Z=VALUE(ROW,COLUMN,LETTER)
      IF(POINTR.EQ.1)NT=Z
      IF(POINTR.EQ.5)ACLEAN=Z
      GOTO 41
44    IF((LETTER.NE.48).AND.(LETTER.NE.49))GOTO 42
      JCODE=LETTER-48
      GOTO 41
45    IF((LETTER.NE.65)
     1 .AND.(LETTER.NE.68)
     2 .AND.(LETTER.NE.83))GOTO 42
      DATYPE=LETTER
      GOTO 41
46    IF((LETTER.NE.89).AND.(LETTER.NE.78))GOTO 42
      PLTYPE=LETTER
      GOTO 41
50    CALL DFAULT
      GOTO 1
60    CALL CLS
      WRITE(3,160)
160   FORMAT(//'+',10X,'Enter a Filename (e.g. TEST1/DAT)'
     1 //'+',14X,'or Press (ENTER) to quit')
      ROW=7
      COLUMN=14
      LETTER=95
      CALL DBOUNC
      CALL BLINKR(ROW,COLUMN,LETTER)
      IF(LETTER.LT.65.OR.LETTER.GT.90)GOTO 1
      CALL STRNG(LABEL,ROW,COLUMN,LETTER,NCHAR)
      IF(LABEL(1).EQ.32)GOTO 1
      DO 61 I=1,12
      FILN(I)=LABEL(I)
61    LABEL(I)=32
      CALL OPEN(7,FILN,0)
      IF(POINTR.EQ.7)GOTO 70
      NEW=.TRUE.
      WRITE(7,161,REC=1)N,TCLEAN,T,TLAG,DPSTOP,VRFLO,FREQ,AMPLIT,
     1 SMALQ,TEMPK,CZERO,TCZERO,
     2 ZK2,TZK2,VZK2,MMD1,SG1,MMD2,SG2,
     3 RHOP,RHOBLK,SE,TSE,WR,SR,TSR,ZKR,TZKR,
     4 NT,JCODE,DATYPE,PLTYPE,ACLEAN,HEAD
161   FORMAT(' ',I2,3F6.1,F5.0,F7.4,F4.1,F5.2,
     1 F7.4,F5.0,F6.2,F5.0,
     2 F6.2,F5.0,F7.4,F4.1,F4.2,F4.1,F4.2,2F6.3,2F5.0,
     3 F6.1,2F5.0,F6.2,F5.0,I3,I1,2I2,F4.2,40A1)
      ENDFILE 7
      GOTO 90
70    CALL OPEN(7,FILN,0)
      READ(7,170,REC=1)JUNK,N,TCLEAN,T,TLAG,DPSTOP,VRFLO,FREQ,AMPLIT,
     1 SMALQ,TEMPK,CZERO,TCZERO,
     2 ZK2,TZK2,VZK2,MMD1,SG1,MMD2,SG2,
     3 RHOP,RHOBLK,SE,TSE,WR,SR,TSR,ZKR,TZKR,
     4 NT,JCODE,DATYPE,PLTYPE,ACLEAN,HEAD
170    FORMAT(A1,I2,3F6.1,F5.0,F7.4,F4.1,F5.2,
     1 F7.4,F5.0,F6.2,F5.0,
     2 F6.2,F5.0,F7.4,F4.1,F4.2,F4.1,F4.2,2F6.3,2F5.0,
     3 F6.1,2F5.0,F6.2,F5.0,I3,I1,2I2,F4.2,40A1)
       ENDFILE 7
       IF(.NOT.NEW)CALL PRINTI(FILN)
       NEW=.FALSE.
       GOTO 1
90     IF(TCLEAN.LT.0.05)TCLEAN=T/N*0.5
       IF(TCZERO.LT.1E-5)TCZERO=25.01
       IF(SE.LT.1E-5)SE=350.0
       IF(WR.LT.1E-5)WR=50.0
       IF(TSE.LT.1E-5)TSE=25.01
5      IF(TZK2.LT.1.0)TZK2=25.01
       IF(VZK2.LT.1E-5)VZK2=0.61
       IF(TSR.LT.1E-5)TSR=25.01
       IF(TZKR.LT.1E-5)TZKR=25.01
       CALL PRINTI(FILN)
       TEMPK=TEMPK+273.0
       TCZERO=TCZERO+273.0
       TSE=TSE+273.0
       TZKR=TZKR+273.0
       TZK2=TZK2+273.0
       TSR=TSR+273.0
       M=4+JCODE*4
       CALL CHECK1(ERROR)
       IF(ERROR)GOTO 70
       CALL SET1
       CALL CHECK2(ERROR)
       GOTO 70
80     CONTINUE
       END
       SUBROUTINE PRINTI(FILN)
       DIMENSION HEAD(40)
       REAL MMD1,MMD2
       INTEGER HEAD,DATYPE,PLTYPE
       LOGICAL HCOPY,FILN(13),DATE(8),TIME(8)
       COMMON /DESIGN/N,T,TCLEAN,TLAG,VRFLO,DPSTOP,FREQ,AMPLIT
       COMMON /TITLE/HEAD
       COMMON /OPDATA/SMALQ,TEMPK,CZERO,TCZERO
       COMMON /FABDUS/ZK2,SE,WR,SR,ZKR,WSTAR
       COMMON /K2EST/TZK2,VZK2,MMD1,SG1,MMD2,SG2,RHOP,RHOBLK
       COMMON /MUCORR/TSE,TSR,TZKR
       COMMON /ACURAC/JCODE
       COMMON /EXTERN/NT,M,WSTART,ACLEAN
       COMMON /OUTPUT/DATYPE,PLTYPE,HCOPY
       HCOPY=.FALSE.
       CALL CLS
       WRITE(3,100)
100    FORMAT('+WOULD YOU LIKE A HARDCOPY PRINTOUT ?')
1      I=2
       J=35
       L=63
       CALL DBOUNC
```

```
      CALL BLINKR(I,J,L)
      IF(L.EQ.78.OR.L.EQ.110)RETURN
      IF(L.EQ.89.OR.L.EQ.121)GOTO 2
      WRITE(3,101)
101   FORMAT(/'+PLEASE ANSWER Y OR N')
      GOTO 1
2     HCOPY=.TRUE.
      CALL DBOUNC
      IF((PEEK(14312).AND.240).EQ.48)GOTO 3
      WRITE(3,102)
102   FORMAT('+PRESS ENTER WHEN LINE PRINTER IS READY')
      I=INKEY(J)
3     CALL DATIME(DATE,TIME)
      WRITE(2,200)FILN,HEAD,DATE,TIME
200   FORMAT('+',80('*')//'+Summary of Input Data For
     1 Baghouse Analysis',10X,'Filename: ',13A1//'+',80('*')//'+',40A1
     2,20X,'Date ',8A1/'+',60X,'Time ',8A1)
      WRITE(2,201)N,TCLEAN,T
201   FORMAT(//'+Basic Design Data'/
     1'+',5X,'Number of Compartments',34X,I2
     2/'+',5X,'Compartment Cleaning Time',29X,F6.1,'    min'
     3/'+',7X,'(Off Line Time)'
     4/'+',5X,'Cleaning Cycle Time',35X,F6.1,'    min')
      IF(TLAG.GT.0.05)WRITE(2,202)TLAG
202   FORMAT('+',5X,'Time Between Cleaning Cycles',26X,F6.1,'    min')
      IF(DPSTOP.GT.0.05)WRITE(2,203)DPSTOP
203   FORMAT('+',5X,'Limiting Pressure Drop',32X,F5.0,'     N/m2')
      IF(TLAG.LT.0.05.AND.DPSTOP.LT.0.05)WRITE(2,204)
204   FORMAT('+',5X,'Continuously Cleaned System')
      WRITE(2,205)VRFLO
205   FORMAT('+',5X,'Reverse Flow Velocity',35X,F7.4,' m/min')
      IF(FREQ.GT.0.05)WRITE(2,206)FREQ
206   FORMAT('+',5X,'Shaking Frequency',39X,F4.1,'     CYCLES/SEC')
      IF(AMPLIT.GT.0.05)WRITE(2,207)AMPLIT
207   FORMAT('+',5X,'Shaking Amplitude',39X,F5.2,'    cm')
      WRITE(2,208)SMALQ,TEMPK,CZERO,TCZERO
208   FORMAT(//'+Operating Data'
     1/'+',5X,'Average Face Velocity',35X,F7.4,' m/min'
     2/'+',5X,'Gas Temperature',39X,F5.0,'      C'
     3/'+',5X,'Inlet Dust Concentration',31X,F6.2,'    g/m3'
     4/'+',10X,'measured at T=',35X,F5.0,'      C')
      WRITE(2,216)
216   FORMAT(//'+Fabric and Dust Properties')
      IF(ZK2.GT.1E-5)GOTO 4
      WRITE(2,209)MMD2,SG2,RHOP,RHOBLK
209   FORMAT('+',5X,'Specific Resistance, K2 Estimated from'
     1/'+',10X,'Mass Median Diameter',31X,F4.1,'    microns'
     2/'+',10X,'Standard Deviation',34X,F4.2
     3/'+',10X,'Particle Density',35X,F6.3,'  g/cm3'
     4/'+',10X,'Bulk Density',39X,F6.3,'  g/cm3')
      GOTO 5
4     CONTINUE
      WRITE(2,210)ZK2,TZK2,VZK2
210   FORMAT('+',5X,'Specific Resistance, K2',32X,F6.2,'   N-min/g-m'
     1/'+',10X,'measured at     T=',31X,F5.0,'      C'
```

Reproduced from best available copy.

```
37900       2  /'+',10X,'                   V=',33X,F7.4,' m/min')
38000          IF(MMD2.LT.1E-5)GOTO 5
38100          WRITE(2,211)MMD1,SG1,MMD2,SG2
38200  211     FORMAT('+',23X,'MMD1=',33X,F4.1,'    microns'
38300       1  /'+',24X,'SG1',35X,F4.2
38400       2  /'+',10X,'Corrected to MMD2=',33X,F4.1,'    microns'
38500       3  /'+',24X,'SG2',35X,F4.2)
38600  5       CONTINUE
38700          WRITE(2,212)SE,TSE,WR
38800  212     FORMAT('+',5X,'Effective Residual Drag, Se',27X,F5.0,
38900       1  '      N-min/m3'/'+',10X,'measured at T=',35X,F5.0,'      C'
39000       2  /'+',5X,'Residual Loading, Wr',34X,F6.1,'    g/m2')
39100          IF(SR.LT.1E-5.AND.ZKR.LT.1E-5)GOTO 6
39200          WRITE(2,213)SR,TSR,ZKR,TZKR
39300  213     FORMAT('+',5X,'Residual Drag, Sr',37X,F5.0,'      N-min/m3'
39400       1  /'+',10X,'measured at T=',35X,F5.0,'      C'
39500       2  /'+',5X,'Initial Slope, Kr',38X,F6.2,'   N-min/g-m'
39600       3  /'+',10X,'measured at T=',35X,F5.0,'      C')
39700  6       WRITE(2,214)NT,JCODE,DATYPE,PLTYPE
39800  214     FORMAT(//'+Special Program Instructions'
39900       1  /'+',5X,'Max Number of Cycles Modelled',27X,I2
40000       2  /'+',5X,'Accuracy Level',43X,I1
40100       3  /'+',5X,'Type of Results Desired',34X,A1
40110       4  /'+',5X,'Plot ?',51X,A1)
40200          IF(ACLEAN.GT.1E-5)WRITE(2,215)ACLEAN
40300  215     FORMAT('+',5X,'Fractional Area Cleaned, Ac',30X,F4.2)
40400          WRITE(2,217)
40500  217     FORMAT(//)
40600          RETURN
40700          END
40800          SUBROUTINE DBOUNC
40900          L=0
41000          DO 1 I=1,2000
41100  1       L=L+1
41200          RETURN
41300          END
41400          SUBROUTINE STRNG(LABEL,ROW,COLUMN,LETTER,NCHAR)
41500          DIMENSION LABEL(40)
41600          INTEGER ROW,COLUMN,ADDRSS
41610          DO 1 I=1,40
41620  1       LABEL(I)=32
41630          LABEL(1)=LETTER
41700          ADDRSS=15360+(ROW-1)*64+COLUMN
41800          CALL POKE(ADDRSS,LETTER)
41900          NCHAR=1
42000          COLUMN=COLUMN+1
42100          ADDRSS=ADDRSS+1
42200  8       LETTER=95
42300          CALL DBOUNC
42400          CALL BLINKR(ROW,COLUMN,LETTER)
42500          IF(LETTER .EQ. 8)GOTO 10
42600          IF(LETTER .EQ. 13)RETURN
42700          NCHAR=NCHAR+1
42800          LABEL(NCHAR)=LETTER
42900          CALL POKE(ADDRSS,LETTER)
```

```
      ADDRSS=ADDRSS+1
      COLUMN=COLUMN+1
      IF(NCHAR .EQ. 40)RETURN
      GOTO 8
10    IF(NCHAR .EQ. 0)GOTO 8
      LABEL(NCHAR)=32
      NCHAR=NCHAR-1
      ADDRSS=ADDRSS-1
      COLUMN=COLUMN-1
      GOTO 8
      END
      SUBROUTINE SETUP(RSTART)
      REAL MMD1,MMD2
      LOGICAL RSTART
      COMMON /DESIGN/N,T,TCLEAN,TLAG,VRFLO,DPSTOP,FREQ,AMPLIT
      COMMON /OPDATA/SMALQ,TEMPK,CZERO,TCZERO
      COMMON /FABDUS/ZK2,SE,WR,SR,ZKR,WSTAR
      COMMON /K2EST/TZK2,VZK2,MMD1,SG1,MMD2,SG2,RHOP,RHOBLK
      COMMON /MUCORR/TSE,TSR,TZKR
      COMMON /EXTERN/NT,M,WSTART,ACLEAN
      COMMON /CALC/DELT,NAREA,IAREA
C-VISCOSITY CORRECTIONS
      VISC(TEMP)=1.46E-3*TEMP**1.5/(TEMP+110.)
C-CHECK FOR RESTART OF CALCULATIONS
      IF(RSTART)GOTO 40
C-DELT
      DELT=T/M/N
      TTEST1=N*TCLEAN+N*TCLEAN*1E-4
      TTEST2=N*TCLEAN-N*TCLEAN*1E-4
      IF(T.GE.TTEST2.AND.T.LE.TTEST1)TCLEAN=TCLEAN-0.0012
      ISKIP=0
      IF(ACLEAN.GT.1E-5)ISKIP=1
C-CALCULATE K2 IF NECESSARY
C     IF K2=0 THEN CALCULATE
C     IFK2)0 AND MMD2=0 DON'T CALC
C     IF K2)0 AND MMD2)0 CORRECT FOR MMD & SIGMA G
      IF(ZK2.GT.1E-5.AND.MMD2.LT.1E-5)GOTO 30
      IF(ZK2.GT.1E-5.AND.MMD2.GT.1E-5)GOTO 20
C CALCULATE K2
      SOLID=RHOBLK/RHOP
      R=(3.0+2.0*SOLID**(5.0/3.0))/(3.0-4.5*SOLID**(1./3.)+
     1 4.5*SOLID**(5./3.)-3.*SOLID**2)
      SO2=36.*10.**(2.304*(ALOG10(SG2))**2)/MMD2**2
C PARTICLE SIZE IN MICRONS, DENSITY IN G/CC, VISCOSITY IN CP
      ZK2=16.64*0.018*R*SO2/6.0/RHOP
      GOTO 30
C CORRECT FOR MMD AND SIGMA G
20    SOB2=36.0*10.0**(2.304*(ALOG10(SG1))**2)/MMD1**2
      SOF2=36.0*10.0**(2.304*(ALOG10(SG2))**2)/MMD2**2
      ZK2=ZK2*SOF2/SOB2
C CORRECT TO VELOCITY OF 0.61 M/MIN
30    ZK2=ZK2*SQRT(0.61/VZK2)
C CORRECT FOR TEMPERATURE
      ZK2=ZK2*VISC(TEMPK)/VISC(TZK2)
      SE=SE*VISC(TEMPK)/VISC(TSE)
      SR=SR*VISC(TEMPK)/VISC(TSR)
      ZKR=ZKR*VISC(TEMPK)/VISC(TZKR)
      CZERO=CZERO*TCZERO/TEMPK
C CORRECT SE T O WR
      SE=SE+WR*ZK2
C CALCULATE WSTART
C INITIAL LOADING ON EACH COMPARTMENT AT TIME ZERO
      IF(DPSTOP.LT.1.0)GOTO 40
      WSTART=(DPSTOP-SE*SMALQ)/(ZK2*SQRT(SMALQ/0.61))/SMALQ+WR
      GOTO 50
```

Reproduced from best available copy.

```
40       WSTART=166.0*(CZERO*SMALQ*(T+TLAG))**0.284-CZERO*SMALQ*T*N
     1   /(N-1)/2.0+WR
50       IF(WSTART.LT.WR)WSTART=WR
C CALCULATE SYSTEM CONSTANT WSTAR (W*)
         WSTAR=0.0
         IF(ZKR.GT.1E-5.AND.SR.GT.1E-5)WSTAR=(SE-SR)/(ZKR-ZK2)
         IF(ISKIP.NE.1)CALL CLEAN(0.0,DUMMY)
C-TOTAL NUMBER OF AREAS ON A BAG (IAREA) AND # TO BE CLEANED (NAREA)
         ERR=0.01
7        I=1.0/ACLEAN+0.3+0.2
         J=1
         IF(ERR.GT.0.06)GOTO 9
         DO 8 I=1,10
         DO 8 J=1,I
         ATEST=FLOAT(J)/FLOAT(I)
         IF(ATEST.LE.(ACLEAN+ERR).AND.ATEST.GE.(ACLEAN-ERR))GOTO 9
8        CONTINUE
         ERR=ERR+0.01
         GOTO 7
9        NAREA=J
         IAREA=I
         RETURN
         END
         SUBROUTINE OUTFIL
         INTEGER HEAD(40)
         LOGICAL OUTPUT,FF
         COMMON /OPDATA/SMALQ,TEMPK,CZERO,TCZERO
         COMMON /FABDUS/ZK2,SE,WR,SR,ZKR,WSTAR
         COMMON /EXTERN/NT,M,WSTART,ACLEAN
         COMMON /CALC/DELT,NAREA,IAREA
         COMMON /TITLE/HEAD
         OUTPUT=3
         FF=13
         CALL CLS
1        WRITE(OUTPUT,100)CZERO,ZK2,SE
         IF(ZKR.GT.1E-5)WRITE(OUTPUT,610)ZKR
         IF(SR.GT.1E-5)WRITE(OUTPUT,620)SR
         WRITE(OUTPUT,600)ACLEAN,DELT,WSTAR
100      FORMAT('+',23X,'Corrected Values'
     1   //'+Inlet Dust Concentration',18X,F6.2,' g/m3'
     2   /'+Corrected to operating temperature'
     3   //'+Fabric and dust cake properties corrected for gas viscosity'
     5   /'+',5X,'Specific cake resistance,K2',10X,F6.2,' N-min/g-m'
     6   /'+',5X,'Effective drag, SE',18X,F5.0,'   N-min/m3')
600      FORMAT('+Fractional area cleaned, AC',17X,F4.2
     1   /'+Time increment',29X,F5.2,' min'
     2   /'+System constant W*',24X,F5.1,'  g/m2'/)
610      FORMAT('+',5X,'Initial Slope, KR',20X,F6.2,' N-min/g-m')
620      FORMAT('+',5X,'Residual Drag, SR',19X,F5.0,' N-min/m3')
         IF(OUTPUT.EQ.2)WRITE(2,2)FF
2        FORMAT('+',1A1)
         IF(OUTPUT.EQ.2)RETURN
         OUTPUT=2
         GOTO 1
         END
```

```
      SUBROUTINE CLEAN(WTOTAL,ACLN)
      COMMON /DESIGN/N,T,TCLEAN,TLAG,VRFLO,DPSTOP,FREQ,AMPLIT
      COMMON /OPDATA/SMALQ,TEMPK,CZERO,TCZERO
      COMMON /FABDUS/ZK2,SE,WR,SR,ZKR,WSTAR
      COMMON /EXTERN/NT,M,WSTART,ACLEAN
      IF(WTOTAL.GT.1.0)GOTO 30
      IF(DPSTOP.LT.1.0)GOTO 20
      WP=(DPSTOP-SE*SMALQ)/(ZK2*SQRT(SMALQ/0.61))/SMALQ+WR
      WPRIME=WP+T*CZERO*SMALQ/2.0
      ACLEAN=1.51E-8*WPRIME**2.52
      IF(FREQ.GT.1E-5)ACLEAN=2.23E-12*(FREQ**2*AMPLIT*WPRIME)**2.52
      GOTO 25
 20   ACLEAN=0.006*(CZERO*SMALQ*(T+TLAG))**0.716
      IF(FREQ.GT.1E-5)ACLEAN=4.9E-4*(FREQ**2*AMPLIT*CZERO*SMALQ*
     1(T+TLAG))**0.716
 25   IF(ACLEAN.GT.1.0)ACLEAN=1.0
      IF(ACLEAN.LT.0.1)ACLEAN=0.1
      RETURN
 30   ACLN=1.51E-8*WTOTAL**2.52
      IF(FREQ.GT.1E-5)ACLN=2.23E-12*(FREQ**2*AMPLIT*WTOTAL)**2.52
      IF(ACLN.GT.1.0)ACLN=1.0
      IF(ACLN.LT.0.1)ACLN=0.1
      RETURN
      END
      SUBROUTINE STABLE(DROP,TIME,JCODE,CONVRG)
      DIMENSION T(50),DP(50),DPDP(20)
      REAL NCHK3,LIM1,LIM2
      LOGICAL CONVRG
      COMMON/STABLD/T1,T2,T001,T002,DP1,DP2,P01,P02,SIGN1,I,N,NL,NCHG
      I=I+1
      T(I)=TIME
      DP(I)=DROP
      T1=T1+TIME
      T2=T1+TIME*TIME
      DP1=DP1+DROP
      DP2=DP2+DROP*TIME
      N=N+1
      IF(I.NE.1)GOTO 45
C SET LIMITS OF CONVERGENCE
      LIM1=0.01
      LIM2=0.001
      LIM1=LIM1/FLOAT(JCODE)
      LIM2=LIM2/FLOAT(JCODE)
      RETURN
 45   TAVG=(T(I)+T(I-1))/2.0
      DELDP=(DP(I)-DP(I-1))/(T(I)-T(I-1))
      IF(I.LT.5)GOTO 10
      N=N-1
      T1=T1-T(I-4)
      T2=T2-T(I-4)*T(I-4)
      DP1=DP1-DP(I-4)
      DP2=DP2-T(I-4)*DP(I-4)
 10   ADELDP=ABS(DELDP)
      IF(ADELDP.LT.1E-20)GOTO 15
      T001=T001+TAVG
```

```
16500           T002=T002+TAVG*TAVG
16600           P01=P01+ALOG(ADELDP)
16700           P02=P02+ALOG(ADELDP)*TAVG
16800           NL=NL+1
16900           IF(NL.LE.3)GOTO 55
17000           B1=(NL*P02-T001*P01)/(NL*T002-T001*T001)
17100     C CHECK # 1
17200           CHK1=EXP(B1*TAVG)
17300           IF(CHK1.LE.LIM1)GOTO 50
17400     15    IF(I.LT.5)GOTO 55
17500           B1=(N*DP2-T1*DP1)/(N*T2-T1*T1)
17600     C CHECK # 2
17700           CHK2=B1/(DROP/(T(I)-T(I-1)))
17800           ACHK2=ABS(CHK2)
17900           IF(ACHK2.GT.LIM2)GOTO 55
18000     50    CONVRG=.TRUE.
18100           RETURN
18200     55    IF((SIGN1*DELDP).GE.0.0.AND.DELDP.NE.0.0)GOTO 20
18300           NCHG=NCHG+1
18400           DPDP(NCHG)=DROP
18500     20    SIGN1=DELDP
18600           IF(NCHG.LT.3)RETURN
18700     C CHECK # 3
18800           NCHK3=ABS(DPDP(NCHG)-DPDP(NCHG-1))-ABS(DPDP(NCHG-1)-
18900          1DPDP(NCHG-2))
19000           IF(NCHK3.LE.0.0)GOTO 50
19200           RETURN
19300           END
```

```
Page 1
        SUBROUTINE BLINKR(ROW,COLUMN,FLASHR)
        INTEGER ROW,COLUMN,FLASHR,ADDRSS,INKEY
        CALL KBINIT
        ADDRSS=15360+(ROW-1)*64+COLUMN
        CALL POKE(ADDRSS,128)
        KOUNT=1
2       LETTER=INKEY(0)
        IF(LETTER)6,3,6
3       LETTER=MOD(KOUNT,20)
        IF(LETTER)5,4,5
4       K=PEEK(ADDRSS)
        IF(K .AND. FLASHR)CALL POKE(ADDRSS,128)
        IF(K .AND. -128)CALL POKE(ADDRSS,FLASHR)
5       KOUNT=KOUNT+1
        IF(KOUNT .GT. 1000)KOUNT=1
        GOTO 2
6       CONTINUE
        FLASHR=LETTER
        CALL POKE(ADDRSS,128)
        RETURN
        END
        INTEGER FUNCTION INKEY(NSCAN)
        DIMENSION KYBRD(8,7,2)
        INTEGER ISHIFT,ADDRSS,START,ROW,COLUMN,SHIFT
        COMMON /ASCKEY/KYBRD
        INKEY=0
        START=X'3800'
        COLUMN=0
        ROW=0
10      SHIFT=1
        DO 1 I=1,7
        ADDRSS=START+2**(I-1)
        INPUT=PEEK(ADDRSS)
        IF(PEEK(X'3880') .OR. 0)SHIFT=2
        IF(INPUT .OR. 0)COLUMN=ISHIFT(INPUT)
        ROW=I
        IF(INPUT .OR. 0)I=7
1       CONTINUE
        IF((NSCAN .EQ. 0) .AND. (COLUMN .EQ. 0))RETURN
        IF(COLUMN .EQ. 0)GOTO 10
        INKEY=KYBRD(COLUMN,ROW,SHIFT)
        RETURN
        END
        INTEGER FUNCTION ISHIFT(I)
        L=.FALSE.
        L=L .OR. 1
        DO 1 J=1,7
        IF(I .AND. L)ISHIFT=J
1       L=L*2
        IF(I .AND. -128)ISHIFT=8
        RETURN
        END
        SUBROUTINE MENU(ELMNTS,POINTR,SFLAG,ROW,COLUMN,LETTER)
        INTEGER ELMNTS,ROW,COLUMN,POINTR,SFLAG
        LOGICAL SFLAG(10)
1       KDASH=15592+64*(POINTR-1)
        IF(SFLAG(POINTR))KDASH=KDASH-25
        KARROW=KDASH+1
        CALL POKE(KDASH,45)
        CALL POKE(KARROW,62)
        CALL DBOUNC
        ROW=POINTR+3
        COLUMN=41
        IF(SFLAG(POINTR))COLUMN=16
```

```
06500          LETTER=62
06600          CALL BLINKR(ROW,COLUMN,LETTER)
06700          CALL POKE(KDASH,128)
06800          IF(LETTER .EQ. 10)LETTER=10
06900          IF((LETTER .NE. 10) .AND. (LETTER .NE. 91))RETURN
07000          CALL POKE(KARROW,128)
07100          I=POINTR
07200          IF((LETTER .EQ. 10) .AND. (I .LT. ELMNTS))POINTR=I+1
07300          IF((LETTER .EQ. 10) .AND. (I .EQ. ELMNTS))POINTR=1
07400          IF((LETTER .EQ. 91) .AND. (I .EQ. 1))POINTR=ELMNTS
07500          IF((LETTER .EQ. 91) .AND. (I .GT. 1))POINTR=I-1
07600          GOTO 1
07700          END
07800          FUNCTION VALUE(ROW,COLUMN,LETTER)
07900          DIMENSION NUMINP(11)
08000          INTEGER ROW,COLUMN,ADDRSS,FLAG
08010          DO 3 I=1,11
08020   3      NUMINP(I)=32
08060          VALUE=0.0
08100          NUMINP(1)=LETTER
08200          ADDRSS=15360+(ROW-1)*64+COLUMN
08300          CALL POKE(ADDRSS,LETTER)
08400          NCHAR=1
08500   1      FLAG=0
08600          CALL DBOUNC
08700          COLUMN=COLUMN+1
08800          ADDRSS=ADDRSS+1
08900   8      LETTER=95
09000          CALL BLINKR(ROW,COLUMN,LETTER)
09100          IF(LETTER.EQ.8)GOTO 10
09200          IF(((LETTER.GE.48).AND.(LETTER.LE.57))
09300         1 .OR.(LETTER.EQ.45)
09400         2 .OR.(LETTER.EQ.46)
09500         3 .OR.(LETTER.EQ.69))FLAG=1
09600          IF(LETTER.EQ.13)GOTO 2
09700          IF(FLAG.EQ.0)GOTO 8
09800          NCHAR=NCHAR+1
09900          NUMINP(NCHAR)=LETTER
10000          CALL POKE(ADDRSS,LETTER)
10050          IF(NCHAR.EQ.11)GOTO 2
10100          GOTO 1
10200   10     IF(NCHAR.EQ.0)GOTO 8
10250          NUMINP(NCHAR)=48
10300          NCHAR=NCHAR-1
10400          ADDRSS=ADDRSS-2
10500          COLUMN=COLUMN-2
10600          GOTO 1
10650   2      IF(NCHAR.LT.11)NUMINP(NCHAR+1)=46
10700          DECODE(NUMINP,20)VALUE
10800   20     FORMAT(E11.4)
10900          RETURN
11000          END
16900          SUBROUTINE KBINIT
17000          DIMENSION KYBRD(8,7,2)
17100          COMMON /ASCKEY/KYBRD
17200          LETTER=63
17300          DO 1 I=1,2
17400          DO 1 J=1,4
17500          DO 1 K=1,8
17600   1      KYBRD(K,J,I)=LETTER+K+(J-1)*8+(I-1)*32
17700          LETTER=47
17800          DO 2 I=1,2
17900          DO 2 J=5,6
18000          DO 2 K=1,8
18100   2      KYBRD(K,J,I)=LETTER+K+(J-5)*8-(I-1)*16
18200          DO 3 K=5,8
```

```
18300          KYCHNG=KYBRD(K,6,1)
18400          KYBRD(K,6,1)=KYBRD(K,6,2)
18500    3     KYBRD(K,6,2)=KYCHNG
18600          KYBRD(1,7,1)=13
18700          KYBRD(2,7,1)=31
18800          KYBRD(3,7,1)=1
18900          KYBRD(4,7,1)=91
19000          KYBRD(5,7,1)=10
19100          KYBRD(6,7,1)=8
19200          KYBRD(7,7,1)=9
19300          KYBRD(8,7,1)=32
19400          KYBRD(1,7,2)=13
19500          KYBRD(2,7,2)=31
19600          KYBRD(3,7,2)=1
19700          KYBRD(4,7,2)=27
19800          KYBRD(5,7,2)=26
19900          KYBRD(6,7,2)=24
20000          KYBRD(7,7,2)=25
20100          KYBRD(8,7,2)=32
20200          RETURN
20300          END
```

```
Page 1
      SUBROUTINE CAKDRG(WDEL,VEL,CDRAG)
      COMMON /FABDUS/ZK2,SE,WR,SR,ZKR,WSTAR
      ZK2V=ZK2*SQRT(VEL*1.6405)
      IF(WSTAR.GT.1E-20)GOTO 10
C-LINEAR MODEL
      CDRAG=ZK2V*(WDEL-WR)
      RETURN
10    WPRIME=WDEL-WR
      EXPO=-1.0*WPRIME/WSTAR
      IF(EXPO.LT.-30.0)EXPO=-30.0
C-NONLINEAR MODEL
      CDRAG=ZK2V*WPRIME+(ZKR-ZK2V)*WSTAR*(1.0-EXP(EXPO))
      RETURN
      END
      SUBROUTINE PENET(CZERO,WEIGHT,VEL,WR,PEN)
      CS=0.0005
      A=400.
      IF(VEL.GT.1E-9)A=0.416/(VEL*3.281)**4+0.094
      IF(VEL.LT.1E-9)VEL=0.0
      XF=1.5E-7
      IF(VEL.GT.1E-9)XF=XF*EXP(12.7*(1.0-EXP(VEL/(-3.2)*3.281)))
      EXPO=(WEIGHT-WR)*A
      PEN=0.0
      IF(EXPO.LT.40.0)PEN=(0.1-XF)*EXP(-EXPO)
      PEN=PEN+XF+CS/CZERO
      RETURN
      END
      SUBROUTINE INITAL
      DIMENSION ZEROM(13),ZEROS(9),IZEROM(4),IZEROS(4)
      COMMON/MODELD/PAVR,TCONT,DTLAST,PENTOT,PAVTOT,DPAVG,QAVG,TLAST,
     1TDSUM,PNMAX,DELP,DPMAX,TREF,IFBAG,NFLAG,JFLAG,LOPCNT
      COMMON/STABLD/T1,T2,T001,T002,DP1,DP2,P01,P02,SIGN1,I,N,NL,NCHG
      EQUIVALENCE (PAVR,ZEROM(1)),(IFBAG,IZEROM(1)),
     1(T1,ZEROS(1)),(I,IZEROS(1))
      DO 10 J=1,4
      IZEROS(J)=0
10    IZEROM(J)=0
      DO 20 J=1,9
20    ZEROS(J)=0.0
      DO 30 J=1,13
30    ZEROM(J)=0.0
      RETURN
      END
      SUBROUTINE RESTRT
      LOGICAL RSTART
      COMMON/EXTERN/NT,M,WSTART,ACLEAN
      COMMON/DESIGN/N,T,TCLEAN,TLAG,VRFLO,DPSTOP,FREQ,AMPLIT
      RSTART=.TRUE.
      DPSTOP=0.0
      CALL SETUP(RSTART)
      WRITE(3,600)ACLEAN
600   FORMAT('+',63('=')
     1/'+',10X,'LIMITING PRESSURE SYSTEM HAS BECOME CONTINUOUS'
     2//'+',10X,'SIMULATION HAS BEEN RESTARTED'
     3/'+',10X,'CALCULATIONS NOW BASED ON CONTINUOUSLY CLEANED
     4SYSTEM'
     5///'+',10X,'REVISED FRACTIONAL AREA CLEANED, AC= ',F4.2
     6//'+',63('='))
      RETURN
      END
      SUBROUTINE CYCLE
```

```
         WRITE(3,10)
 10      FORMAT('+',10X,'PRESS ANY KEY TO CONTINUE')
         J=1
         I=INKEY(J)
         RETURN
         END
         SUBROUTINE PRINTO(OUTPUT)
         LOGICAL OUTPUT,HCOPY
         COMMON /OUTPUT/I,J,HCOPY
         WRITE(3,10)
 10      FORMAT('+PRESS ANY KEY TO CONTINUE')
         I=INKEY(1)
         IF(HCOPY)OUTPUT=2
         RETURN
         END
```

```
Page 1
               SUBROUTINE FINIT(FILL)
               LOGICAL FILL(81),BLANK,PERIOD,BAR
               DATA BLANK,PERIOD,BAR/' ','.','I'/
               DO 10 J=1,81
         10    FILL(J)=BLANK
               FILL(1)=BAR
               FILL(21)=PERIOD
               FILL(41)=PERIOD
               FILL(61)=PERIOD
               FILL(81)=PERIOD
               RETURN
               END
```

```
Page 1
00100          PROGRAM FMODEL
00200          DIMENSION HEAD(40)
00300          REAL MMD1,MMD2
00400          INTEGER HEAD,DATYPE,PLTYPE
00500          LOGICAL ERROR,ALDIAG,PLDIAG,PRDIAG,NEW,HCOPY,RSTART,FILN(13)
00550          LOGICAL DATE(8),TIME(8),JUNK
00600          COMMON /DESIGN/N,T,TCLEAN,TLAG,VRFLO,DPSTOP,FREQ,AMPLIT
00700          COMMON /TITLE/HEAD
00800          COMMON /OPDATA/SMALQ,TEMPK,CZERO,TCZERO
00900          COMMON /FABDUS/ZK2,SE,WR,SR,ZKR,WSTAR
01000          COMMON /K2EST/TZK2,VZK2,MMD1,SG1,MMD2,SG2,RHOP,RHOBLK
01100          COMMON /MUCORR/TSE,TSR,TZKR
01200          COMMON /ACURAC/JCODE
01300          COMMON /EXTERN/NT,M,WSTART,ACLEAN
01400          COMMON /OUTPUT/DATYPE,PLTYPE,HCOPY
01500          COMMON /DIAG/ALDIAG,PRDIAG,PLDIAG
01600          RSTART=.FALSE.
01700          HCOPY=.TRUE.
01800          CALL CLS
01900          WRITE(3,5)
02000    5     FORMAT(' ENTER THE NAME OF THE INPUT FILE ')
02100          READ(3,10)FILN
02200    10    FORMAT(13A1)
02300          CALL OPEN(7,FILN,0)
02400          READ(7,20)JUNK,N,TCLEAN,T,TLAG,DPSTOP,VRFLO,FREQ,AMPLIT,
02500         1 SMALQ,TEMPK,CZERO,TCZERO,
02600         2 ZK2,TZK2,VZK2,MMD1,SG1,MMD2,SG2,
02700         3 RHOP,RHOBLK,SE,TSE,WR,SR,TSR,ZKR,TZKR,
02800         5 NT,JCODE,DATYPE,PLTYPE,ACLEAN,HEAD
02900          ENDFILE 7
03000          CALL MASSAG
03100          CALL SETUP(RSTART)
03210          CALL DATIME(DATE,TIME)
03220          CALL CLS
03230          WRITE(3,30)FILN,HEAD,DATE,TIME
03240    30    FORMAT(
03250         1 '+',63('*')
03260         2 /'+',3X,'Results of Baghouse Analysis',10X,'Filename: ',13A1
03270         3 /'+',63('*')
03280         4 /'+',40A1,5X,'Date ',8A1/'+',45X,'Time ',8A1)
03290          IF(PEEK(14312).EQ.63)GOTO 40
03300          WRITE(3,50)
03310    50    FORMAT('+Press (ENTER) when printer is ready')
03320          I=INKEY(1)
03330    40    WRITE(2,30)FILN,HEAD,DATE,TIME
03340          CALL OUTFIL
03400          CALL MODEL
03700    20    FORMAT(A1,I2,3F6.1,F5.0,F7.4,F4.1,F5.2,
03800         1 F7.4,F5.0,F6.2,F5.0,
03900         2 F6.2,F5.0,F7.4,F4.1,F4.2,F4.1,F4.2,2F6.3,2F5.0,
04000         3 F6.1,2F5.0,F6.2,F5.0,I3,I1,2I2,F4.2,40A1)
04100          END
05600          SUBROUTINE MASSAG
05700          DIMENSION HEAD(40)
05800          REAL MMD1,MMD2
05900          INTEGER HEAD,POINTR,ELMNTS,ROW,COLUMN,DATYPE,PLTYPE
06000          LOGICAL ALDIAG,PLDIAG,PRDIAG
06100          LOGICAL HCOPY
06200          COMMON /DESIGN/N,T,TCLEAN,TLAG,VRFLO,DPSTOP,FREQ,AMPLIT
06300          COMMON /TITLE/HEAD
06400          COMMON /OPDATA/SMALQ,TEMPK,CZERO,TCZERO
06500          COMMON /FABDUS/ZK2,SE,WR,SR,ZKR,WSTAR
06600          COMMON /K2EST/TZK2,VZK2,MMD1,SG1,MMD2,SG2,RHOP,RHOBLK
```

```
      COMMON /MUCORR/TSE,TSR,TZKR
      COMMON /ACURAC/JCODE
      COMMON /EXTERN/NT,M,WSTART,ACLEAN
      COMMON /OUTPUT/DATYPE,PLTYPE,HCOPY
      COMMON /DIAG/ALDIAG,PRDIAG,PLDIAG
      DATA ALDIAG,PRDIAG,PLDIAG/3*.FALSE./
   80 IF(TCLEAN.LT.0.05)TCLEAN=T/N*0.5
      IF(TCZERO.LT.1E-5)TCZERO=25.01
      IF(SE.LT.1E-5)SE=350.0
      IF(WR.LT.1E-5)WR=50.0
      IF(TSE.LT.1E-5)TSE=25.01
    5 IF(TZK2.LT.1.0)TZK2=25.01
      IF(VZK2.LT.1E-5)VZK2=0.61
      IF(TSR.LT.1E-5)TSR=25.01
      IF(TZKR.LT.1E-5)TZKR=25.01
      TEMPK=TEMPK+273.0
      TCZERO=TCZERO+273.0
      TSE=TSE+273.0
      TZKR=TZKR+273.0
      TZK2=TZK2+273.0
      TSR=TSR+273.0
      M=4+JCODE*4
      JCODE=1+JCODE*2
      IF(PLTYPE.EQ.89)PLDIAG=.TRUE.
      IF(DATYPE.EQ.65)GOTO 92
      IF(DATYPE.EQ.68)ALDIAG=.TRUE.
      PRDIAG=.TRUE.
   92 CONTINUE
      END
```

```
Page 1
      SUBROUTINE MODEL
      DIMENSION HEAD(40)
      DIMENSION TIME(30),OLDTIM(30),CAKE(30),SBAG(30),QBAG(30)
      DIMENSION WD(10,30),S(10,30),QAREA(10),P(10)
      INTEGER HEAD,UNIT
      LOGICAL PLDIAG,PRDIAG,LDIAG,HCOPY,CONVRG,DATE(8),SKIP
      LOGICAL TI(8),BUFFER(81),BLANK,ASTER,PLUS,BAR,CIRCLE,DOT,AA,BB
      COMMON /DESIGN/N,T,TCLEAN,TLAG,VRFLO,DPSTOP,FREQ,AMPLIT
      COMMON /TITLE/HEAD
      COMMON /OPDATA/SMALQ,TEMPK,CZERO,TCZERO
      COMMON /FABDUS/ZK2,SE,WR,SR,ZKR,WSTAR
      COMMON /ACURAC/JCODE
      COMMON /DIAG/LDIAG,PRDIAG,PLDIAG
      COMMON /EXTERN/NT,M,WSTART,ACLEAN
      COMMON /OUTPUT/I,J,HCOPY
      COMMON /CALC/DELT,NAREA,IAREA
      COMMON /MODELD/PAVR,TCONT,DTLAST,PENTOT,PAVTOT,DPAVG,QAVG,TLAST,
     1 TDSUM,PNMAX,DELP,DPMAX,TREF,IFBAG,NFLAG,JFLAG,LOPCNT
      DATA DRAG,BAG1,BAG2/'AREA','SBAG','QBAG'/
      DATA BLANK,ASTER,PLUS,BAR/' ','*','+','I'/
      DATA CIRCLE,DOT,AA,BB/'o','.',27,20/
      UNIT=3
      IF(.NOT.HCOPY)GOTO 20
      UNIT=2
      GOTO 20
 10   CZERO=CZEROE
      CALL RESTRT
C
C     Zero all of the parameters in the MODELD and STABLD COMMON blocks
C
 20   CALL INITAL
      CONVRG=.FALSE.
      AREA=1.0/IAREA
      DO 30 I=1,IAREA
      QAREA(I)=SMALQ
      DO 30 IBAG=1,N
      OLDTIM(IBAG)=-2
      TIME(IBAG)=-1
      S(IAREA,IBAG)=0.0
 30   WD(I,IBAG)=WSTART
      CZEROE=CZERO
      TLAG=IFIX(TLAG/DELT+0.4)*DELT
      IF(TLAG.LT.0.01)TLAG=0.0
      TMOD=TLAG+T
      IF(DPSTOP.GT.0.0)TMOD=1E+20
C Specify the drag through the fabric
C
      SFAB=SR
      IF(WSTAR.LT.1E-20)SFAB=SE
      DO 40 NDIS=1,2
 40   CALL FILL(NDIS)
      NDIS=1
      PLMAX=1500.
      PLMIN=0.
      TMN=T/M/N
C
C     Beginning of time loop
C
      DO 270 JLOOP=1,20000
      CALL SHIFT(1)
      X=(DELP-PLMIN)/(PLMAX-PLMIN)
      NPT=X*40
      CALL PTPOK(NPT,1)
```

```
06410            IF(PEEK(X'3820').EQ.8)NDIS=2
06420            IF(PEEK(X'3820').EQ.32)NDIS=1
06500            CALL DISPLY(NDIS)
06600            DELT=TMN
06700            TTEST=AMOD(TCONT+0.01,TMOD)-0.01
06800            IF(TTEST.LT.0.005)TTEST=0
06900            IF(TTEST.GT.(T-0.005).AND.TTEST.LT.(T+0.005))TTEST=T
07000            IF(TCONT.LT.1E-9.OR.TTEST.LE.-0.01.OR.TTEST.GE.0.01)GOTO 100
07100      C
07200      C     A cycle has been completed
07300      C
07310            LOPCNT=LOPCNT+1
07320      C
07400      C     Compute average values for cycle
07500      C
07800            Q1=QSYSTM*DTLAST
07900            P1=PENTOT*DTLAST
08000            D1=DELP*DTLAST
08100            T1=2.0*TDSUM
08200            QAVGN=(QAVG-Q1)/T1
08300            PAVNOW=(PAVTOT-P1)/T1
08400            DPAVGN=(DPAVG-D1)/T1
08500            TDSUM=0.0
08550            TDIF=TCONT-TLAST
08600            TLAST=TCONT
08700            QAVG=Q1
08800            PAVTOT=P1
08900            DPAVG=D1
09000            IF(CONVRG)GOTO 290
09100      C
09200      C     PLMAX, PTMAX, and PLMIN are used for
09300      C     plotting only.
09400      C
09500            PLMAX=DPMAX
09600            PTMAX=PNMAX
09700      C
09800      C     TREF is used for printing results only.
09900      C
10000            TREF=TCONT
10100      C
10200      C     Reset the maximum values of delta P and penetration
10300      C
10400            PNMAX=0.0
10500            DPMAX=0.0
10600      C
10700      C Check for a limiting pressure system for forced continuous operation
10800      C Check for time between cleaning cycles equal to zero
10900      C
11000            CONTST=TDIF-T
11100            IF(DPSTOP.GT.0.5.AND.LOPCNT.GT.3.AND.
11200          1 CONTST.GT.-TMN.AND.CONTST.LE.(TMN+0.01))GOTO 10
11300      C
11400      C     Test for convergence
11500      C
11600            CALL STABLE(DPAVGN,TCONT,JCODE,CONVRG)
11610      C
11620      C     Update the convergence display
11630      C
11640            CALL SHIFT(2)
11650            X=DPAVGN/1500.
11660            NPT=X*40
11670            CALL PTPOK(NPT,2)
11700      C
11800      C     Test for the number of cycles to be modelled
11900      C
12000            LOPTST=NT-LOPCNT
```

```
         IF(LOPTST.LE.1)WRITE(UNIT,80)LOPCNT
80       FORMAT('+*** CONVERGENCE TO STEADY STATE NOT REACHED AFTER ',I3,
      1  ' CYCLES'//)
         IF(LOPTST.LE.1)CONVRG=.TRUE.
C
C        Write out the header for the line printer plot
C
         IF(CONVRG.AND.PLDIAG)WRITE(2,90)AA,BB,PLMAX,PTMAX
90       FORMAT('+',2A1//'+',22X,'Pressure',4X,'Fractional',6X,'0',34X,
      1  'Delta P (o)',25X,1PE9.2/'+',9X,'Time',11X,'Drop',6X,
      2  'Penetration',5X,'0',32X,'Penetration (+)',23X,1PE9.2
      3  /'+',9X,'(min)',9X,'(N/m2)',21X,80('-'))
C
C        Continue with new cycle
C
100      CONTINUE
         IF(TTEST.GT.T)GOTO 120
C
C Extra pass for cleaned bag
C
C        Bag loop   1
C        Increment time and goto 130 if a bag was just cleaned
C
         DO 110 IFBAG=1,N
         IF(OLDTIM(IFBAG).LE.TIME(IFBAG))GOTO 110
         TCONT=TCONT+0.01
         GOTO 130
110      CONTINUE
C
C        End of bag loop 1
C
120      IFBAG=0
         DELT=TMN
C
C Determine time
C
         TCONT=(JLOOP-1)*DELT
130      TTEST=AMOD(TCONT+0.01,TMOD)-0.01
         IF(TTEST.LT.0.005)TTEST=0
         IF(TTEST.GT.(T-0.005).AND.TTEST.LT.(T+0.005))TTEST=T
C        WRITE(2,400)TTEST,TCONT
400      FORMAT('+ TTEST= ',F6.2,' TCONT = ',F6.2)
C
C        Bag loop   2
C        Calculate Area, Compartment, and System Drags
C
         SSYSTM=0.0
         DELTT=DELT
         VRFLOW=0.0
         CLEAN=T-TCLEAN-0.001
C
         DO 190 IBAG=1,N
         SBAG(IBAG)=0.0
C
C        Loop on areas
C
         DO 150 I=1,IAREA
C
C If a bag was just cleaned, estimate new drag so as to
C compute a new flow velocity.
C
         IF(S(IAREA,IBAG).LT.1E+19)GOTO 140
         CALL CAKDRG(WD(I,IBAG),SMALQ,S(I,IBAG))
         S(I,IBAG)=S(I,IBAG)+SFAB
140      CONTINUE
         IF(TCONT.GT.1E-9)QAREA(I)=DELP/S(I,IBAG)
```

```
C
C Determine drag on each area
C
        CALL CAKDRG(WD(I,IBAG),QAREA(I),S(I,IBAG))
        S(I,IBAG)=S(I,IBAG)+SFAB
150     SBAG(IBAG)=SBAG(IBAG)+AREA/S(I,IBAG)
C
C       End of area loop
C
C       Compute compartment drag.
C
        SBAG(IBAG)=1.0/SBAG(IBAG)
C
C  Determine time in cycle
C
        IF(TTEST.GT.(T+0.005))GOTO 180
        OLDTIM(IBAG)=TIME(IBAG)
        TIME(IBAG)=AMOD(TTEST+0.01+IBAG*T/N,T)-0.01
C       WRITE(2,410)IBAG,OLDTIM(IBAG),TIME(IBAG)
410     FORMAT('+ IBAG = ',I3,' OLDTIM = ',F6.2,' TIME = ',F6.2)
C
C Test for an off line
C
        IF(TIME(IBAG).GE.CLEAN)GOTO 160
        IF(TCONT.LT.1E-9)GOTO 180
        IF(TIME(IBAG).GT.0.005)GOTO 180
        IF(TTEST.LE.0.01.AND.TLAG.GT.1E-9)GOTO 180
C
C       Compartment # IBAG is being cleaned, so set its drag
C       very high (i.e. there is no flow through it)
C
160     DO 170 I=1,IAREA
170     S(I,IBAG)=1E20
        SBAG(IBAG)=1E20
C       WRITE(2,420)IBAG
420     FORMAT('+ CLEANING COMPARTMENT # ',I3)
        VRFLOW=VRFLO
C
C       Add this compartment drag to system drag
C
180     SSYSTM=SSYSTM+1.0/SBAG(IBAG)
        IF(OLDTIM(IBAG).GT.TIME(IBAG).AND.TTEST.LT.(T+0.005))DELTT=0.01
C       IF(OLDTIM(IBAG).GT.TIME(IBAG).AND.TTEST.LT.(T+0.005))TIME(IBAG)=
C     1 -1.0
190      CONTINUE
C
C       End of bag loop 2
C
C
C       Calculate system drag, pressure drop and flow velocity
C
        SSYSTM=1.0/SSYSTM
        DELP=SMALQ*SSYSTM*N+VRFLOW*SSYSTM
        QSYSTM=SMALQ+VRFLOW/N
C
C Correct inlet concentration for reverse flow air
C
        CZERO=CZEROE*(QSYSTM-VRFLOW/N)/QSYSTM
C
C       Bag loop 3
C       Calculate flows and penetrations.  Clean a bag if necessary
C
        PENTOT=0.0
        SKIP=.FALSE.
        IF(TTEST.GT.T)SKIP=.TRUE.
C
```

```
             DO 240 IBAG=1,N
             IF(SKIP)GOTO 200
             DELT=DELTT
             IF((TIME(IBAG)+TMN).GT.(T-TCLEAN))DELT=T-TCLEAN-TIME(IBAG)
      200    CONTINUE
             IF(SBAG(IBAG).GT.1E19)GOTO 215
             CAKE(IBAG)=0.0
      C
      C Determine the penetration and new dust loadings for
      C each area of this compartment
      C
             DO 210 I=1,IAREA
             QAREA(I)=DELP/S(I,IBAG)
             CALL PENET(CZERO,WD(I,IBAG),QAREA(I),WR,P(I))
             WAREA=QAREA(I)*(1.-P(I))*DELT*CZERO
             CAKE(IBAG)=CAKE(IBAG)+WD(I,IBAG)*AREA
             PENTOT=PENTOT+P(I)*AREA*QAREA(I)/QSYSTM/N
      210    WD(I,IBAG)=WD(I,IBAG)+WAREA
      C      WRITE(2,440)IBAG,DELT
      440    FORMAT('+ COMPARTMENT # ',I3,' DELT = ',1PE11.4)
      C      WRITE(2,450)(P(I),I=1,IAREA),PENTOT
      450    FORMAT('+',12(1PE11.4))
      C
      C      End of area loop
      C
      C      Compute flow through this compartment
      C
      215    QBAG(IBAG)=DELP/SBAG(IBAG)
             IF(SKIP)GOTO 240
             IF(OLDTIM(IBAG).LE.TIME(IBAG))GOTO 240
      C
      C This loop is entered when a compartment is about to
      C come back on line.  It computes the amount of mass
      C removed from the bags in the compartment and sets the
      C
             WDUMP=0.0
      C
      C There are NAREA areas which must be cleaned.
      C
             DO 230 II=1,NAREA
             WMAX=0.0
             DO 220 I=1,IAREA
             IF(WD(I,IBAG).LT.WMAX)GOTO 220
             WMAX=WD(I,IBAG)
             IFAREA=I
      220    CONTINUE
      C
      C Compute the amount of dust to be removed from this area
      C
             WDUMP=WDUMP+(WD(IFAREA,IBAG)-WR)*AREA
      C
      C Dump the dust from this area
      C
      230    WD(IFAREA,IBAG)=WR
      C
      C
      C
      C
      C End of bag loop 3
      C
      240    CONTINUE
      C
      C      Compute average values for this time increment
      C
             DELT=DELTT
             D1=DTLAST+DELT
```

```
        DPAVG=DPAVG+D1*DELP
        QAVG=QAVG+D1*QSYSTM
        PAVTOT=PAVTOT+PENTOT*D1
        PAVR=PAVR+PENTOT*D1
        TDSUM=TDSUM+DTLAST
        DTLAST=DELT
        CONTOT=PENTOT*CZERO
C
C       Find the maximum pressure drop and penetration for this cycle
C
        IF(PENTOT.GT.PNMAX)PNMAX=PENTOT
        IF(DELP.GT.DPMAX)DPMAX=DELP
        IF(DELP.LT.DPMIN)DPMIN=DELP
        IF(.NOT.CONVRG)GOTO 260
        IF(.NOT.PLDIAG.AND..NOT.PRDIAG)GOTO 260
C
C       Plot the pressure drop and penetration as a function of time
C
        PT=TCONT-TREF
        CALL FINIT(BUFFER)
        K3=INT(DELP/PLMAX*80.0)+1
        IF(K3.GT.81)K3=81
        BUFFER(K3)=CIRCLE
        K3=INT(PENTOT/PTMAX*80.0)+1
        IF(K3.GT.81)K3=81
        BUFFER(K3)=PLUS
        WRITE(2,250)PT,DELP,PENTOT,(BUFFER(K3),K3=1,81)
250     FORMAT('+',3X,F10.2,5X,F10.0,5X,1PE9.2,8X,81A1)
C
C       Loop
C
260     IF(IFBAG.NE.0)GOTO 120
        IF(DPSTOP.LT.1E-9)GOTO 270
        IF(TMOD.LT.1.E19.AND.TTEST.GT.(T+TMN))TMOD=TCONT-T-TMN
        IF(TTEST.LE.T.OR.DELP.LT.DPSTOP)GOTO 270
        TMOD=TCONT
270     CONTINUE
C
C End of time loop
C
280     FORMAT(/'+','For',F10.2,' minutes operation,
     1   Cycle number ',I3
     2  /'+',10X,'Average Penetration=',15X,1PE9.2
     3  /'+',10X,'Average Pressure Drop=',8X,0PF10.2,'    N/m2'
     4  /'+',10X,'Average System Flow=',12X,0PF10.4,' m/min'
     5  /'+',10X,'Maximum Penetration=',15X,1PE9.2
     6  /'+',10X,'Maximum Pressure Drop=',8X,0PF10.2,'    N/m2'
     7  //)
C
C       End of modelling
C
290     CONTINUE
        WRITE(UNIT,280)TDIF,LOPCNT,PAVNOW,DPAVGN,QAVGN,PNMAX,DPMAX
        CALL DATIME(DATE,TI)
        WRITE(UNIT,300)TI
300     FORMAT('+End at ',8A1//)
        RETURN
        END
```

```
Page 1
              SUBROUTINE DFAULT
              REAL MMD1,MMD2
              INTEGER DATYPE,HEAD(40),PLTYPE
              LOGICAL ALDIAG,PLDIAG,PRDIAG,HCOPY
              COMMON /DESIGN/N,T,TCLEAN,TLAG,VRFLO,DPSTOP,FREQ,AMPLIT
              COMMON /OPDATA/SMALQ,TEMPK,CZERO,TCZERO
              COMMON /FABDUS/ZK2,SE,WR,SR,ZKR,WSTAR
              COMMON /K2EST/TZK2,VZK2,MMD1,SG1,MMD2,SG2,RHOP,RHOBLK
              COMMON /MUCORR/TSE,TSR,TZKR
              COMMON /ACURAC/JCODE
              COMMON /DIAG/ALDIAG,PRDIAG,PLDIAG
              COMMON /EXTERN/NT,M,WSTART,ACLEAN
              COMMON /OUTPUT/DATYPE,PLTYPE,HCOPY
              COMMON /TITLE/HEAD
              DO 1 I=1,40
    1         HEAD(I)=32
              N=0
              T=00.0
              TCLEAN=0.0
              TLAG=00.0
              DPSTOP=0.0
              VRFLO=0.0
              FREQ=0.0
              AMPLIT=0.0
              SMALQ=0.00
              TEMPK=000.0
              CZERO=00.0
              TCZERO=25.01
              ZK2=0.00
              TZK2=25.01
              VZK2=0.61
              MMD1=0.0
              MMD2=0.0
              SG1=0.0
              SG2=0.0
              RHOP=0.0
              RHOBLK=0.0
              SE=350.0
              TSE=25.01
              WR=50.0
              SR=0.0
              TSR=25.01
              ZKR=0.0
              TZKR=25.01
              NT=00
              JCODE=0
              DATYPE=65
              PLTYPE=78
              ACLEAN=0.0
              RETURN
              END
              SUBROUTINE DISPLY
              RETURN
              END
              SUBROUTINE CLS
              LOGICAL A,B
              A=X'1C'
              B=X'1F'
              WRITE(3,10)A,B
    10        FORMAT('+',2A1)
              RETURN
              END
              ERROR=.TRUE.
              WRITE(OUTPUT,660)
      660     FORMAT('+',10X,'MMD OF DUST OUT OF RANGE 2 TO 50 MICRONS')
      140     IF(SG2.GE.2..AND.SG2.LE.4.)GOTO 180
              ERROR=.TRUE.
              WRITE(OUTPUT,670)
      670     FORMAT('+',10X,'STANDARD DEVIATION OF DUST OUT OF RANGE 2 TO 4')
      680     FORMAT('+',10X,'BULK DENSITY EXCEEDS DISCRETE PARTICLE DENSITY')
      690     FORMAT('+',10X,'BULK OR DISCRETE DENSITY MISSING')
      610     FORMAT('+',10X,'BOTH TIMED AND PRESSURE CONTROLLED CLEANINGS'
            1 /'+',10X,'SPECIFIED - ONLY ONE IS VALID')
      500     FORMAT(
            1 '+',63('*')
            2 /'+',22X,'DIAGNOSTIC MESSAGES'
            3 /'+',63('*')/'+',40A1)
      510     FORMAT('+',10X,'NUMBER OF COMPARTMENTS MUST RANGE FROM 2 TO 30')
      520     FORMAT('+',10X,'THE NUMBER OF COMPARTMENTS TIMES THE CLEANING'
            1 /'+',10X,'CYCLE TIME MUST BE LESS THAN TOTAL CLEANING TIME')
      530     FORMAT('+',10X,'THE COMPARTMENT CLEANING TIME MUST BE LESS THAN'
            1 /'+',10X,'THE TOTAL CYCLE TIME')
      540     FORMAT('+',10X,'A GAS TEMPERATURE HAS NOT BEEN ENTERED')
      560     FORMAT('+',10X,'TIME INCREMENT TOO SMALL, I.E. ( 0.01 MINUTES')
      570     FORMAT('+',10X,'INVALID FREQUENCY ORR AMPLITUDE FOR SHAKER')
      180     IF(.NOT.ERROR)RETURN
              IF(OUTPUT.EQ.2)WRITE(2,2)FF
      2       FORMAT('+',1A1)
              IF(OUTPUT.EQ.2)RETURN
              CALL PRINTO(OUTPUT)
              IF(OUTPUT.NE.2)RETURN
              GOTO 1
              END
              SUBROUTINE SET1
              REAL MMD1,MMD2
              COMMON /DESIGN/N,T,TCLEAN,TLAG,VRFLO,DPSTOP,FREQ,AMPLIT
              COMMON /OPDATA/SMALQ,TEMPK,CZERO,TCZERO
              COMMON /FABDUS/ZK2,SE,WR,SR,ZKR,WSTAR
              COMMON /K2EST/TZK2,VZK2,MMD1,SG1,MMD2,SG2,RHOP,RHOBLK
              COMMON /MUCORR/TSE,TSR,TZKR
              COMMON /EXTERN/NT,M,WSTART,ACLEAN
              COMMON /CALC/DELT,NAREA,IAREA
      C-VISCOSITY CORRECTIONS
              VISC(TEMP)=1.46E-3*TEMP**1.5/(TEMP+110.)
      C-CALCULATE K2 IF NECESSARY
      C       IF K2=0 THEN CALCULATE
      C       IFK2)0 AND MMD2=0 DON'T CALC
      C       IF K2)0 AND MMD2)0 CORRECT FOR MMD & SIGMA G
              IF(ZK2.GT.1E-5.AND.MMD2.LT.1E-5)GOTO 30
              IF(ZK2.GT.1E-5.AND.MMD2.GT.1E-5)GOTO 20
      C CALCULATE K2
              SOLID=RHOBLK/RHOP
              R=(3.0+2.0*SOLID**(5.0/3.0))/(3.0-4.5*SOLID**(1./3.)+
            1 4.5*SOLID**(5./3.)-3.*SOLID**2)
              SO2=36.*10.**(2.304*(ALOG10(SG2))**2)/MMD2**2
      C PARTICLE SIZE IN MICRONS, DENSITY IN G/CC, VISCOSITY IN CP
              ZK2=16.64*0.018*R*SO2/6.0/RHOP
              GOTO 30
      C CORRECT FOR MMD AND SIGMA G
      20      SOB2=36.0*10.0**(2.304*(ALOG10(SG1))**2)/MMD1**2
              SOF2=36.0*10.0**(2.304*(ALOG10(SG2))**2)/MMD2**2
              ZK2=ZK2*SOF2/SOB2
      C CORRECT TO VELOCITY OF 0.61 M/MIN
      30      ZK2=ZK2*SQRT(0.61/VZK2)
      C CORRECT FOR TEMPERATURE
              ZK2=ZK2*VISC(TEMPK)/VISC(TZK2)
              SE=SE*VISC(TEMPK)/VISC(TSE)
              SR=SR*VISC(TEMPK)/VISC(TSR)
```

```
        ZKR=ZKR*VISC(TEMPK)/VISC(TZKR)
C CORRECT SE TO WR
        SE=SE+WR*ZK2
C CALCULATE SYSTEM CONSTANT WSTAR (W*)
        WSTAR=0.0
        IF(ZKR.GT.1E-5.AND.SR.GT.1E-5)WSTAR=(SE-SR)/(ZKR-ZK2)
        RETURN
        END
        SUBROUTINE CHECK2(ERROR)
        INTEGER HEAD(40)
        LOGICAL ERROR,OUTPUT,FF
        COMMON /TITLE/HEAD
        COMMON /OPDATA/SMALQ,TEMPK,CZERO,TCZERO
        COMMON /FABDUS/ZK2,SE,WR,SR,ZKR,WSTAR
        COMMON /EXTERN/NT,M,WSTART,ACLEAN
        OUTPUT=3
        FF=12
1       IF(WSTAR.LT.1E-5)GOTO 45
        IF(SE.GE.SR)GOTO 46
        ERROR=.TRUE.
        WRITE(OUTPUT,200)
200     FORMAT('+',10X,'EFFECTIVE DRAG, SE, IS LESS THAN RESIDUAL, SR')
45      IF(ZKR.LT.1E-5.AND.SR.LT.1E-5)GOTO 50
        ERROR=.TRUE.
        WRITE(OUTPUT,620)
620     FORMAT('+',10X,'INCOMPLETE DATA FOR NON-LINEAR DRAG MODEL')
46      IF(SR.GT.1E-5)GOTO 47
        ERROR=.TRUE.
        WRITE(OUTPUT,630)
630     FORMAT('+',10X,'RESIDUAL DRAG SR IS MISSING')
47      IF(ZKR.GT.1E-5)GOTO 50
        ERROR=.TRUE.
        WRITE(OUTPUT,640)
640     FORMAT('+',10X,'INITIAL SLOPE KR IS MISSING')
50      IF(ACLEAN.GE.0.0.AND.ACLEAN.LE.1.0)GOTO 60
        ERROR=.TRUE.
        WRITE(OUTPUT,600)
600     FORMAT('+',10X,'FRACTIONAL AREA CLEANED OUT OF RANGE 0 TO 1')
60      TESTK2=ZK2*298**1.5/408.0*(TEMPK+110)/TEMPK**1.5
        IF(TESTK2.GE.0.25.AND.TESTK2.LE.10.0)GOTO 75
        WRITE(OUTPUT,610)
610     FORMAT('+',10X,'K2 IS OUT OF RANGE 0.25 TO 10')
        ERROR=.TRUE.
75      IF(.NOT.ERROR)RETURN
        IF(OUTPUT.EQ.2)WRITE(2,2)FF
2       FORMAT('+',1A1)
        IF(OUTPUT.EQ.2)RETURN
        CALL PRINTO(OUTPUT)
        IF(OUTPUT.NE.2)RETURN
        WRITE(2,100)HEAD
100     FORMAT('+',75('*')/'+',28X,'DIAGNOSTIC MESSAGES'
     1  /'+',75('*'),/'+',40A1)
        GOTO 1
        END
        SUBROUTINE PRINTO(OUTPUT)
        LOGICAL OUTPUT
        WRITE(3,10)
10      FORMAT('+DO YOU WANT HARD COPY PRINTOUT? ')
        I=INKEY(1)
        IF(I.EQ.89.OR.I.EQ.121)OUTPUT=2
        RETURN
        END
```

CHAPTER 5

PLUME OPACITY PREDICTION MODEL

Program Author: Dr. A. S. Damle

REVISION 1.0

ABSTRACT

This program determines the effect of condensation of vapors in stack gases on the opacity of the plume as the stack gases are cooled and diluted by mixing with ambient air. Both heterogeneous condensation and homogeneous nucleation of vapors are considered in the condensation process. The user is allowed to input parameters to include stack gas as well as ambient conditions. The program then calculates plume opacity at selected plume travel time increments.

USER BENEFITS

- To our knowledge, this program is a first attempt to correlate mathematically the increase in opacity of so-called "detached plumes" to the condensation of vapors present in stack gases.
- The program enables a user to determine the increase in opacity of a plume in a variety of scenarios and thus may help to determine an optimum strategy to control plume opacity.

HARDWARE/SOFTWARE REQUIREMENTS

This program runs on a TRS-80 Model I, III, or 4 microcomputer with the NEWDOS/80 Version 2.0 or TRSDOS 1.3 operating system. It requires 48K of RAM and at least one disk drive. A printer is required for the output.

The program "PLUME" and the input program "PLUMIN" are written in Microsoft FORTRAN but reference some subroutines written in ASSEMBLER. If any routines are to be modified, the Microsoft programs F80, M80, and L80 are required.

Since the program uses machine- and ROM-dependent coding, this version will be expected to run only on the TRS-80 machines mentioned above with the TRSDOS or NEWDOS operating systems.

METHOD

This program predicts the plume behavior after it is released from the stack. The plume gases are continually cooled and diluted by mixing with ambient air. The rate of mixing, or the entrainment, depends upon the local plume and ambient properties such as velocity and temperature. A dew point may eventually be reached in the plume leading to an onset of condensation of the vapors. The condensation process and plume dilution change the concentration and size distribution of aerosols in the plume and thus change the plume opacity. Ordinary differential equations are set up to describe the change in each plume variable with plume travel time. These are solved by a 4th order Runge-Kutta method to obtain all plume variables as a function of time. Full details of the modeling equations are discussed by Damle, Ensor, and Sparks (1).

The most critical parameters in the plume opacity prediction appear to be sulfuric acid vapor concentration and the particulate phase concentration and size distribution. Stack gas moisture content and temperature have a moderate effect. The stack gas velocity, ambient wind velocity, stack gas and ambient temperatures, and entrainment parameters affect the plume dilution and cooling rate and thus influence the distance from the stack exit at which the maximum opacity is reached but have a somewhat moderate influence on the magnitude of the peak plume opacity itself. High dilution rates usually predict higher peak plume opacities because the homogeneous nucleation contribution may increase quickly due to an increase in the supersaturation ratio. Low dilution rates allow the vapors more time to condense heterogeneously, thus reducing contribution from homogeneous nucleation.

GENERAL RESTRICTIONS

Although this program will deliver results for almost any inputs, negative values should not be used for any of the input quantities. Reasonable default values are supplied.

This program considers binary condensation of sulfuric acid and water vapor due to the current interest in air pollution. However, the program can easily be adapted to any other vapor system by modifying the GROWTH subroutine in the program. Essentially, vapor-liquid equilibrium data for the new system are required along with the physical properties of the new vapor species for this modification.

To compute the opacity of the plume a refractive index of all the particles was assumed to be 1.5 - 0.05i with a mean light wavelength of 550 nm. For other refractive index and light wavelength parameters a new set of extinction efficiencies, Q, for all size increments should be computed and supplied in the OPACITY subroutine. The above-mentioned values were selected for this program as being suitable for fly ash particles. In this program, the particle size spectrum is divided into 25 increments with a geometric ratio of 2, starting from a 0.01-μm radius.

SAMPLE PROBLEM

Statement: Assume you have a stack gas containing 20 ppm sulfuric acid. Calculate the change in opacity of the stack gas due to condensation.

Results: This sample problem serves two purposes. First, the user can compare his results with the numbers listed below to verify that the program is operating properly. Second, the step-by-step instructions given below will aid the novice in learning how to use the program.

There are two programs that are used to run this model: PLUMIN and PLUME. The PLUMIN program is a menu-driven data entry program that writes data files to be read by PLUME. To use this program, begin with the PLUMIN data entry program. When this program is initiated, the Main Menu will appear as shown in Figure 5-1. From this menu the user can choose between two data entry options, two file options and the exit option. Notice that the user has the option of pressing <H> for help. If this is done, the help pages will be shown on the screen in the sequence shown in Figure 5-2. Press any key to return to the Main Menu.

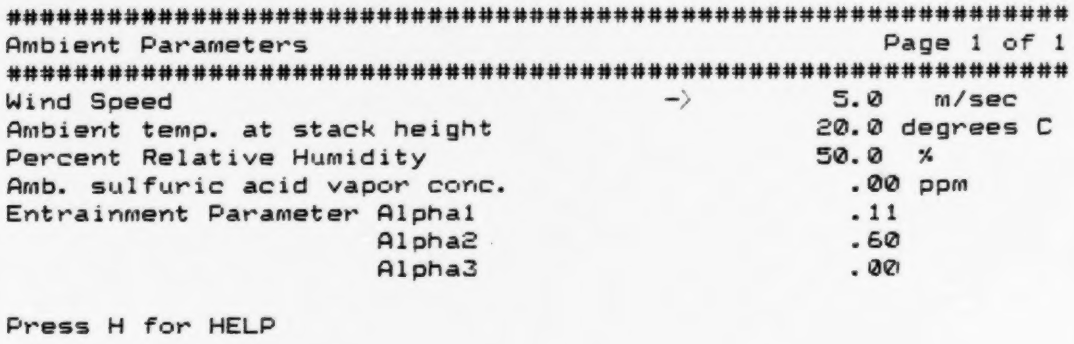
For this sample problem, select the first option, "Input Ambient Parameters," by pressing <ENTER>. The screen will appear as shown in Figure 5-3. The menu title is shown in the upper left corner. To its right, "Page 1 of 1" is displayed. This tells the user that this is the only data page for the "Input Ambient Parameters" option. Pressing <H> will obtain Figure 5-4, the third help page. Read this then return to the "Ambient Parameters" menu. The parameters listed in this menu are self-explanatory with the exception of the last three items (Alpha 1, Alpha 2, and Alpha 3). These

```
**************************************************************
                    RTI Plume model
**************************************************************
            ->      Input Ambient parameters
                    Input Plume parameters
                    Save Input Parameters in a File
                    Read Input Parameters from a File
                    Exit


                    Press H for HELP
```

Figure 5-1. The Main Menu for the PLUMIN data entry program.

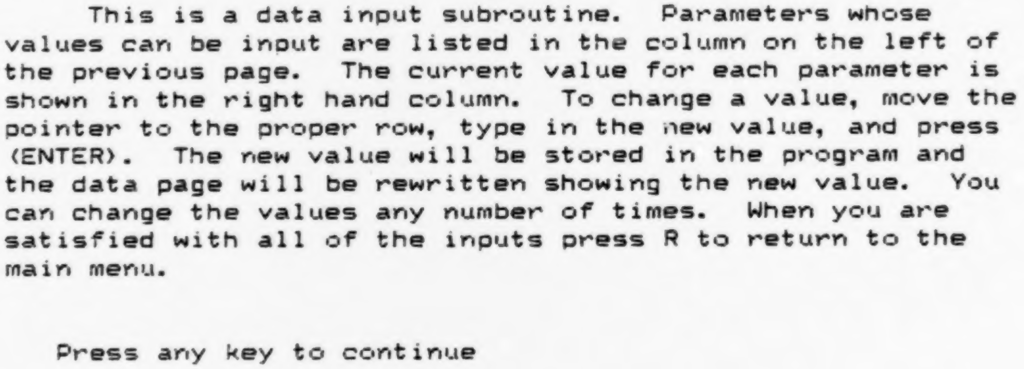
```
        The previous page listed 5 commands available to you :
   2 data inputs; store input parameters in a disk file;
   retrieve input parameters from a disk file; and Exit.
   To execute any of these commands move the pointer to
   the proper row using the up and down arrow keys on the left
   end of the keyboard and select the subroutine by pressing the
   <ENTER> key.  For example, to enter parameters describing
   ambient conditions near the stack, move the pointer to the
   row labelled "Input Ambient parameters" and press the <ENTER>
   key.  A list of variables will be displayed on the screen
   and instructions will be given on how to enter data.


      Press any key to continue



        When the program is first started, the parameters are
   initialized with the values which are typical of plumes.
   You can change any or all of the values in the program.
   When all of the data has been entered it must be saved in a
   data file.  Move the pointer to the row labelled "Save Input
   Parameters in a file" and press <ENTER>.  Enter a filename
   as requested.  The program will write your input values to
   disk and ask you if you want a hardcopy of the parameter
   values.  Control will be returned to the main menu where you
   can enter new values for another data file.  When you are
   finished entering data, use the "Exit" option to end the
   program.
           Press any key to continue
```

Figure 5-2. Help pages for the Main Menu.



```
################################################################
Ambient Parameters                                   Page 1 of 1
################################################################
Wind Speed                               ->        5.0   m/sec
Ambient temp. at stack height                     20.0 degrees C
Percent Relative Humidity                         50.0  %
Amb. sulfuric acid vapor conc.                     .00 ppm
Entrainment Parameter Alpha1                       .11
                      Alpha2                       .60
                      Alpha3                       .00

Press H for HELP
```

Figure 5-3. "Input Ambient Parameters" menu.



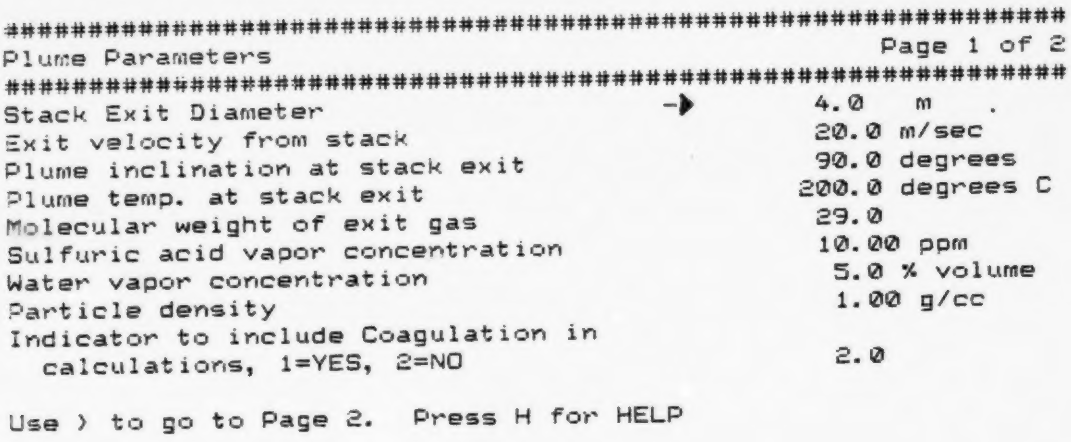
```
     This is a data input subroutine.  Parameters whose
values can be input are listed in the column on the left of
the previous page.  The current value for each parameter is
shown in the right hand column.  To change a value, move the
pointer to the proper row, type in the new value, and press
<ENTER>.  The new value will be stored in the program and
the data page will be rewritten showing the new value.  You
can change the values any number of times.  When you are
satisfied with all of the inputs press R to return to the
main menu.


   Press any key to continue
```

Figure 5-4. Help page for the "Ambient Parameters" menu.

are constants that are used in a semiempirical equation that predicts the plume dilution rate. The equation is discussed by Damle, Ensor and Sparks (1). In most cases, the default values listed in the menu will be adequate. Advanced users may be interested in changing these values for rigorous calculations. Now return to the Main Menu by pressing <R>.

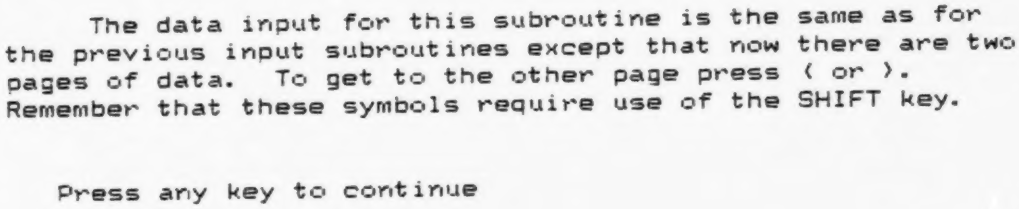
Next, choose the "Input Plume Parameters" option by using the down arrow key to position the pointer, then pressing <ENTER>. This will result in Figure 5-5 appearing on the screen. Reading the upper right corner, "Page 1 of 2," tells the user that this is the first page of the two pages that can be used to enter plume data. Pressing <H> will obtain Figure 5-6, the last of the help pages. Return to the "Input Plume Parameters" menu and change the "Sulfuric acid vapor concentration" value to 20.0 ppm. To do this, position the pointer next to "Sulfuric acid vapor concentration" using the down arrow key. Enter the new value by typing <20><ENTER>. In this menu the parameters are easily understood with the possible exception of the last item (coagulation indicator). This flag tells the modeling program whether or not to include particle coagulation in the calculations. To be rigorous, this flag should be set to YES (i.e., enter a value of 1.0); however, the inclusion of coagulation increases the computation time. Therefore, for practical purposes, this value should be considered as "fine-tuning." The user should set this value to 2.0 (no coagulation) until better accuracy is desired.

As instructed at the bottom of this page, use the > key to go to the second page of this menu, Figure 5-7. This menu allows the user to enter the parameters describing the particle size distribution, which may be bimodal (as has been observed for some utility boilers). The first four parameters in the menu describe the small particle mode and the last four parameters describe the large particle mode. If a bimodal distribution is not desired, change the number (or mass) concentration for mode 2 to zero. For this problem, change the value of "concentration (number)" in "Mode 2" to 5E+3. When using this program it is only neccessary to enter one of the concentration values (either number or mass). When <ENTER> is pressed the other value is automatically recalculated to comply with the new entry. When these two values have been entered, check the screen to make sure that the correct value was typed (using the < allows you to return to page 1 of this menu to check values), then return to the Main Menu.



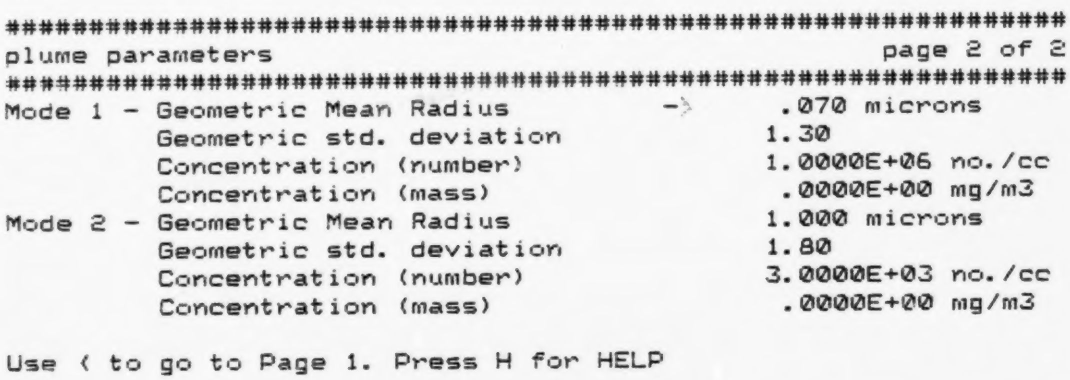
```
################################################################
Plume Parameters                                     Page 1 of 2
################################################################
Stack Exit Diameter                          ->    4.0   m    .
Exit velocity from stack                           20.0 m/sec
Plume inclination at stack exit                    90.0 degrees
Plume temp. at stack exit                         200.0 degrees C
Molecular weight of exit gas                       29.0
Sulfuric acid vapor concentration                  10.00 ppm
Water vapor concentration                           5.0 % volume
Particle density                                    1.00 g/cc
Indicator to include Coagulation in
  calculations, 1=YES, 2=NO                         2.0

Use > to go to Page 2.  Press H for HELP
```

Figure 5-5. Page 1 of the "Input Plume Parameters" menu.



```
      The data input for this subroutine is the same as for
the previous input subroutines except that now there are two
pages of data.  To get to the other page press < or >.
Remember that these symbols require use of the SHIFT key.


   Press any key to continue
```

Figure 5-6. Help page called in "Plume Parameters" menu.

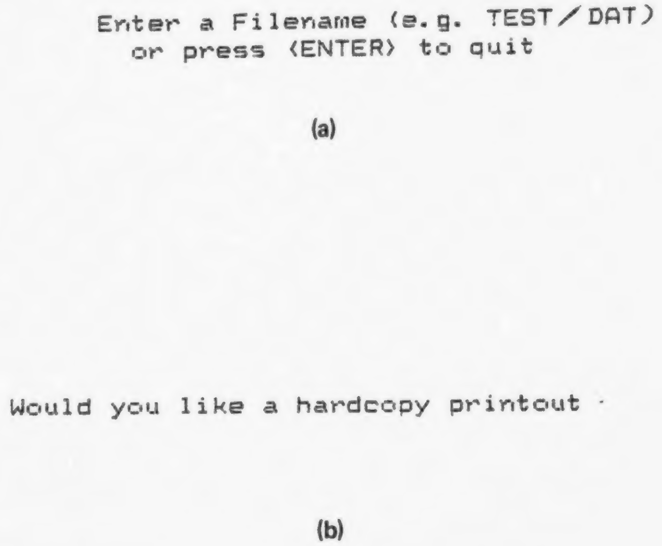


```
################################################################
plume parameters                                     page 2 of 2
################################################################
Mode 1 - Geometric Mean Radius              ->     .070 microns
         Geometric std. deviation                 1.30
         Concentration (number)                   1.0000E+06 no./cc
         Concentration (mass)                      .0000E+00 mg/m3
Mode 2 - Geometric Mean Radius                    1.000 microns
         Geometric std. deviation                 1.80
         Concentration (number)                   3.0000E+03 no./cc
         Concentration (mass)                      .0000E+00 mg/m3

Use < to go to Page 1. Press H for HELP
```

Figure 5-7. Page 2 of the "Input Plume Parameters" menu.

To save the values you have just entered for use with the PLUME program, choose the "Save Input..." option. The program will request a file name as in Figure 5-8a; type TEST/DAT:1 <ENTER>. The program will ask if you desire a printout of the data file. See Figure 5-8b. Type Y to obtain a printout. For this sample problem, the printout should appear as Figure 5-9. If it does not, return to the data entry options and enter the correct values before repeating the "Save.." step. When the printout is correct, the user is ready to exit the program and run the PLUME program. However, before exiting, choose the "Read Input..." option. The screen will again appear as in Figure 5-8a. This option is used to specify an already created file so that the user can use data that have previously been entered as a base for a new run. For example, all data may be the same except "wind speed." To avoid the time-consuming process of re-entering all the data, merely call the old data file and use the data entry options to change the differing value. This new data set can be saved under the old file name (erasing the old values) or under a new name. Now exit the program.

The next step is to initiate the PLUME program. The user should have the printer ready before beginning. The program will ask for the data file name (must be a file created using the "Save..." option of the PLUMIN program), plume travel time, and a time interval between printouts. Plume travel time will usually be in the range of 0 to 10 seconds. Larger values may be used but will increase the length of time needed to run the program. It is recommended that the user begin with a value inside this range and increase the time interval only if the results are insufficient. The time interval between printouts tells the program how often to print out intermediate values. It does not influence the actual calculation. A value of 0.1 will usually give good results, but this can be varied as the user desires. Smaller values will cause printout time to increase. As these questions are put forth on the screen, respond as shown in Figure 5-10. The program will automatically generate the output presented as Figure 5-11. After the calculations have been completed and the results printed, the program will ask for the number of datasets remaining. When more than one file has been created using PLUMIN, the user may enter a number greater than 0, which will cause the program to return to the beginning and request another file name. Type <0> to exit the program.



```
Enter a Filename (e.g. TEST/DAT)
  or press <ENTER> to quit
```

(a)

```
Would you like a hardcopy printout ·
```

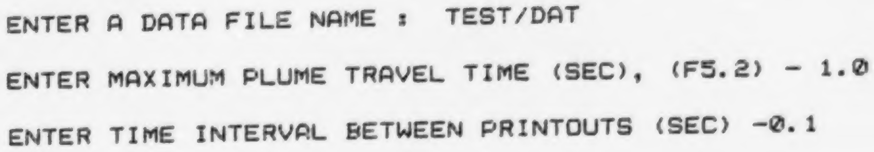
(b)

Figure 5-8. Response to "Save . . ." or "Read" options of PLUMIN.

```
*****************************************************************************

Summary of Input Data for Plume Model          Filename: TEST/DAT

*****************************************************************************

                                                              Date  1/14/85
                                                              Time 16:57:24


Ambient Variables

     Wind Speed                                                 5.0  m/sec
     Ambient temperature                                       20.0  C
     Percent Relative Humidity                                 50.0  %
     Ambient sulfuric acid vapor concentration                  .00 ppm
     Entrainment Parameters,  Alpha1                            .11
                              Alpha2                            .60
                              Alpha3                            .00


Plume Variables

     Stack Exit Diameter                                        4.0  m
     Exit Velocity at stack                                    20.0  m/sec
     Plume inclination at stack exit                           90.0  degrees
     Plume temperature at stack exit                          200.0  degrees C
     Molecular weight of exit gas                              29.0
     Sulfuric acid vapor concentration                         20.00 ppm
     Water vapor concentration                                  5.0  % volume
     Particle density                                           1.00 gm/cm3


 Initial particulate size distribution and concentration

     Mode 1 - Geometric Mean Radius                             .070 microns
              Geometric Std. Deviation                         1.30
              Concentration (number)                           1.0000E+06 /cm3
              Concentration (mass)                              .0000E+00 mg/m3

     Mode 2 - Geometric Mean Radius                            1.000 microns
              Geometric Std. Deviation                         1.80
              Concentration (number)                           5.0000E+03 /cm3
              Concentration (mass)                             9.9158E+01 mg/m3

      Indicator to include Coagulation in
        calculations,  1=YES, 2=NO                              2.0
```

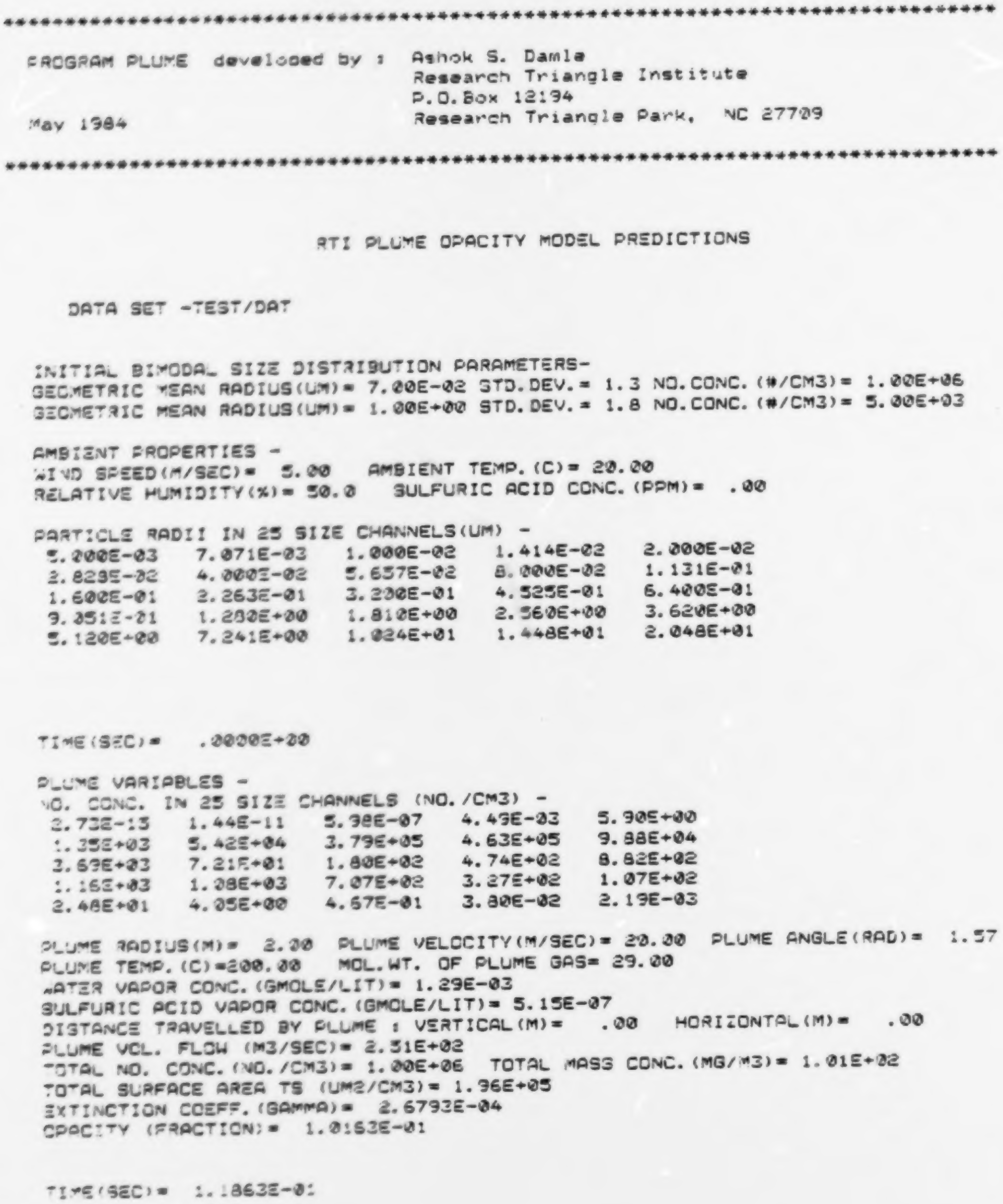
Figure 5-9. Printout for PLUMIN program.

Reproduced from best available copy.



```
ENTER A DATA FILE NAME :  TEST/DAT

ENTER MAXIMUM PLUME TRAVEL TIME (SEC), (F5.2) - 1.0

ENTER TIME INTERVAL BETWEEN PRINTOUTS (SEC) -0.1
```

Figure 5-10. User input to PLUME program.



```
************************************************************************

 PROGRAM PLUME  developed by :  Ashok S. Damla
                                Research Triangle Institute
                                P.O.Box 12194
 May 1984                       Research Triangle Park,  NC 27709

************************************************************************


                     RTI PLUME OPACITY MODEL PREDICTIONS


    DATA SET -TEST/DAT


 INITIAL BIMODAL SIZE DISTRIBUTION PARAMETERS-
 GEOMETRIC MEAN RADIUS(UM)= 7.00E-02 STD.DEV.= 1.3 NO.CONC.(#/CM3)= 1.00E+06
 GEOMETRIC MEAN RADIUS(UM)= 1.00E+00 STD.DEV.= 1.8 NO.CONC.(#/CM3)= 5.00E+03

 AMBIENT PROPERTIES -
 WIND SPEED(M/SEC)=  5.00   AMBIENT TEMP.(C)= 20.00
 RELATIVE HUMIDITY(%)= 50.0   SULFURIC ACID CONC.(PPM)=  .00

 PARTICLE RADII IN 25 SIZE CHANNELS(UM) -
  5.000E-03   7.071E-03   1.000E-02   1.414E-02   2.000E-02
  2.828E-02   4.000E-02   5.657E-02   8.000E-02   1.131E-01
  1.600E-01   2.263E-01   3.200E-01   4.525E-01   6.400E-01
  9.051E-01   1.280E+00   1.810E+00   2.560E+00   3.620E+00
  5.120E+00   7.241E+00   1.024E+01   1.448E+01   2.048E+01



 TIME(SEC)=   .0000E+00

 PLUME VARIABLES -
 NO. CONC. IN 25 SIZE CHANNELS (NO./CM3) -
  2.73E-13   1.44E-11   5.98E-07   4.49E-03   5.90E+00
  1.35E+03   5.42E+04   3.79E+05   4.63E+05   9.88E+04
  3.69E+03   7.21E+01   1.80E+02   4.74E+02   8.82E+02
  1.16E+03   1.08E+03   7.07E+02   3.27E+02   1.07E+02
  2.48E+01   4.05E+00   4.67E-01   3.80E-02   2.19E-03

 PLUME RADIUS(M)=  2.00  PLUME VELOCITY(M/SEC)= 20.00  PLUME ANGLE(RAD)=  1.57
 PLUME TEMP.(C)=200.00   MOL.WT. OF PLUME GAS= 29.00
 WATER VAPOR CONC.(GMOLE/LIT)= 1.29E-03
 SULFURIC ACID VAPOR CONC.(GMOLE/LIT)= 5.15E-07
 DISTANCE TRAVELLED BY PLUME : VERTICAL(M)=   .00   HORIZONTAL(M)=   .00
 PLUME VOL. FLOW (M3/SEC)= 2.31E+02
 TOTAL NO. CONC.(NO./CM3)= 1.00E+06  TOTAL MASS CONC.(MG/M3)= 1.01E+02
 TOTAL SURFACE AREA TS (UM2/CM3)= 1.96E+05
 EXTINCTION COEFF.(GAMMA)=  2.6793E-04
 OPACITY (FRACTION)=  1.0163E-01


 TIME(SEC)=  1.1863E-01
```

Figure 5-11. Output from PLUME program.

193

```
PLUME VARIABLES -
NO. CONC. IN 25 SIZE CHANNELS (NO./CM3) -
 1.80E-15    9.47E-12    3.95E-07    2.97E-03    3.89E+00
 8.93E+02    3.57E+04    2.50E+05    3.25E+05    6.52E+04
 2.43E+03    4.76E+01    1.19E+02    3.12E+02    5.82E+02
 7.65E+02    7.10E+02    4.66E+02    2.16E+02    7.07E+01
 1.63E+01    2.67E+00    3.08E-01    2.51E-02    1.44E-03

PLUME RADIUS(M)=  3.24  PLUME VELOCITY(M/SEC)= 11.55  PLUME ANGLE(RAD)=  1.37
PLUME TEMP.(C)=118.43   MOL.WT. OF PLUME GAS= 28.95
WATER VAPOR CONC.(GMOLE/LIT)= 1.01E-03
SULFURIC ACID VAPOR CONC.(GMOLE/LIT)= 3.40E-07
DISTANCE TRAVELLED BY PLUME : VERTICAL(M)=  1.73   HORIZONTAL(M)=   .17
PLUME VOL. FLOW (M3/SEC)= 3.81E+02
TOTAL NO. CONC.(NO./CM3)= 6.63E+05  TOTAL MASS CONC.(MG/M3)= 6.67E+01
TOTAL SURFACE AREA TS (UM2/CM3)= 1.29E+05
EXTINCTION COEFF.(GAMMA)=  1.7676E-04
OPACITY (FRACTION)=  1.0824E-01


TIME(SEC)=  2.3509E-01

PLUME VARIABLES -
NO. CONC. IN 25 SIZE CHANNELS (NO./CM3) -
 2.23E-06     .00E+00     .00E+00     .00E+00     .00E+00
 8.58E+01    6.90E+03    8.76E+04    2.24E+05    1.57E+05
 3.79E+04    3.61E+03    2.32E+02    2.44E+02    4.56E+02
 6.02E+02    5.59E+02    3.67E+02    1.70E+02    5.56E+01
 1.28E+01    2.09E+00    2.42E-01    1.97E-02    1.13E-03

PLUME RADIUS(M)=  4.13  PLUME VELOCITY(M/SEC)=  9.06  PLUME ANGLE(RAD)=  1.23
PLUME TEMP.(C)= 91.91   MOL.WT. OF PLUME GAS= 28.94
WATER VAPOR CONC.(GMOLE/LIT)= 8.98E-04
SULFURIC ACID VAPOR CONC.(GMOLE/LIT)= 2.54E-07
DISTANCE TRAVELLED BY PLUME : VERTICAL(M)=  2.87   HORIZONTAL(M)=   .48
PLUME VOL. FLOW (M3/SEC)= 4.86E+02
TOTAL NO. CONC.(NO./CM3)= 5.19E+05  TOTAL MASS CONC.(MG/M3)= 5.38E+01
TOTAL SURFACE AREA TS (UM2/CM3)= 1.27E+05
EXTINCTION COEFF.(GAMMA)=  1.7442E-04
OPACITY (FRACTION)=  1.3427E-01


TIME(SEC)=  3.4130E-01

PLUME VARIABLES -
NO. CONC. IN 25 SIZE CHANNELS (NO./CM3) -
 6.00E+06    8.41E+06    3.43E+07    3.94E+07    2.78E+07
 7.78E+06    3.60E+06    5.65E+05    1.24E+05    1.76E+05
 1.29E+05    3.68E+04    4.37E+03    4.13E+02    3.91E+02
 5.14E+02    4.79E+02    3.14E+02    1.45E+02    4.74E+01
 1.09E+01    1.78E+00    2.05E-01    1.67E-02    9.60E-04

PLUME RADIUS(M)=  4.80  PLUME VELOCITY(M/SEC)=  7.94  PLUME ANGLE(RAD)=  1.13
PLUME TEMP.(C)= 78.70   MOL.WT. OF PLUME GAS= 28.93
WATER VAPOR CONC.(GMOLE/LIT)= 8.34E-04
SULFURIC ACID VAPOR CONC.(GMOLE/LIT)= 1.55E-07
DISTANCE TRAVELLED BY PLUME : VERTICAL(M)=  3.70   HORIZONTAL(M)=   .82
PLUME VOL. FLOW (M3/SEC)= 5.74E+02
TOTAL NO. CONC.(NO./CM3)= 1.28E+08  TOTAL MASS CONC.(MG/M3)= 5.38E+01
TOTAL SURFACE AREA TS (UM2/CM3)= 6.27E+05
EXTINCTION COEFF.(GAMMA)=  2.9141E-04
OPACITY (FRACTION)=  2.4401E-01


TIME(SEC)=  4.4398E-01
```

Figure 5-11. Output from PLUME program (continued).

```
PLUME VARIABLES -
NO. CONC. IN 25 SIZE CHANNELS (NO./CM3) -
 1.74E+05    1.30E+06    6.32E+06    1.84E+07    3.54E+07
 3.77E+07    2.20E+07    7.16E+06    1.35E+06    2.72E+05
 1.34E+05    4.31E+04    5.91E+03    5.11E+02    3.47E+02
 4.52E+02    4.21E+02    2.76E+02    1.28E+02    4.16E+01
 9.59E+00    1.56E+00    1.80E-01    1.46E-02    8.42E-04

PLUME RADIUS(M)=  5.35  PLUME VELOCITY(M/SEC)=  7.28  PLUME ANGLE(RAD)=  1.05
PLUME TEMP.(C)= 70.25   MOL.WT. OF PLUME GAS= 28.93
WATER VAPOR CONC.(GMOLE/LIT)= 7.90E-04
SULFURIC ACID VAPOR CONC.(GMOLE/LIT)= 3.70E-09
DISTANCE TRAVELLED BY PLUME : VERTICAL(M)=  4.39   HORIZONTAL(M)=  1.18
PLUME VOL. FLOW (M3/SEC)= 6.55E+02
TOTAL NO. CONC.(NO./CM3)= 1.30E+08  TOTAL MASS CONC.(MG/M3)= 6.48E+01
TOTAL SURFACE AREA TS (UM2/CM3)= 1.62E+06
EXTINCTION COEFF.(GAMMA)=  3.9131E-04
OPACITY (FRACTION)=  3.4219E-01


TIME(SEC)=  4.8576E-01

PLUME VARIABLES -
NO. CONC. IN 25 SIZE CHANNELS (NO./CM3) -
 1.55E+05    1.37E+06    5.86E+06    1.73E+07    3.35E+07
 3.62E+07    2.13E+07    6.99E+06    1.33E+06    2.64E+05
 1.29E+05    4.12E+04    5.67E+03    4.89E+02    3.31E+02
 4.31E+02    4.02E+02    2.63E+02    1.22E+02    3.97E+01
 9.15E+00    1.49E+00    1.72E-01    1.40E-02    8.03E-04

PLUME RADIUS(M)=  5.56  PLUME VELOCITY(M/SEC)=  7.08  PLUME ANGLE(RAD)=  1.02
PLUME TEMP.(C)= 67.55   MOL.WT. OF PLUME GAS= 28.93
WATER VAPOR CONC.(GMOLE/LIT)= 7.76E-04
SULFURIC ACID VAPOR CONC.(GMOLE/LIT)= 1.02E-09
DISTANCE TRAVELLED BY PLUME : VERTICAL(M)=  4.65   HORIZONTAL(M)=  1.33
PLUME VOL. FLOW (M3/SEC)= 6.87E+02
TOTAL NO. CONC.(NO./CM3)= 1.24E+08  TOTAL MASS CONC.(MG/M3)= 6.21E+01
TOTAL SURFACE AREA TS (UM2/CM3)= 1.56E+06
EXTINCTION COEFF.(GAMMA)=  3.7617E-04
OPACITY (FRACTION)=  3.4165E-01
```

Figure 5-11. Output from PLUME program (continued).

This is the end of the sample problem. At this point the user may return to the PLUMIN program and change values in order to examine the response of the PLUME program.

REFERENCES

1. A. S. Damle, D. S. Ensor, L. E. Sparks. Prediction of the Opacity of Detached Plumes Formed by Condensation of Vapors. Atmospheric Environment 18 (2): 435-444. 1984.

```
C
C ************************************************************
C
C     COPYRIGHT (C) 1984 : DR. ASHOK S. DAMLE
C     ALL COMMERCIAL RIGHTS RESERVED
C
C ************************************************************
C
C     PROGRAM PLUME  -  Version 1.0        December 1984
C
C     Developed by      Dr. Ashok S. Damle,
C                       Research Triangle Institute
C                       P.O.Box 12194
C                       Research Triangle Park, NC 27709
C
C                       Phone: (919) 541-5829
C
C ************************************************************
C
C     This program predicts change in opacity of a plume due to
C     condensation of vapors present in the stack gases.
C
C
      PROGRAM PLUME
      DIMENSION A(34),F(34),R(25),V(25),CPA(25)
      LOGICAL FILN(13),CONTRL
      REAL MWA,MWP,MWPO
      DATA  MWA/28.9/,R1/0.005/,CPA/25*0.0/
      DATA  N/25/,NT/34/,DTOLD/0.1/
C
C     Setup 25 particle size channels and compute
C     their respective size and volume
C
      R(1)=R1
      DLNR=ALOG(2.0)/2.0
      DO 1 I=2,N
      R(I)=EXP((I-1)*DLNR+ALOG(R(1)))
 1    V(I)=4.189*R(I)**3
      V(1)=4.189*R(1)**3
C
C     Input data from data file, and also provide
C     maximum plume travel time and printing interval
C
      CALL CLS
      WRITE(3,190)
 190  FORMAT(/'+',60('*') //'+',1X,'PROGRAM PLUME  developed by :',
     1 '  Ashok S. Damle'/'+',1X,'Version 1.0',20X,'Research Triangle
     2 Institute'/'+',32X,'P.O.Box 12194'/'+',1X,'December
     3 1984',18X,'Research Triangle Park, NC 27709'//'+',17X, 'Copyright

     4 (C) 1984  Dr. Ashok S. Damle'/'+',17X,'All Commercial
     5 Rights Rreserved'//'+',60('*')//)
 173  WRITE(3,170)
 170  FORMAT(' ENTER A DATA FILE NAME ')
      DO 174 I=1,13
 174  FILN(I)=32
      READ(3,171)FILN
 171  FORMAT(13A1)
      WRITE(3,181)
 181  FORMAT(/' ENTER MAXIMUM PLUME TRAVEL TIME (SEC), (F5.2) - ')
      READ(3,182)TEND
 182  FORMAT(F5.2)
      WRITE(3,183)
 183  FORMAT(/' ENTER TIME INTERVAL BETWEEN PRINTOUTS (SEC) - ')
      READ(3,182)DTPR
      CALL OPEN(7,FILN,0)
      READ(7,172,REC=1)CONTRL,UA,TA,RH,CHAP,AL1,AL2,AL3,DPO,UO ,DEGO,
     1 TPO,MWPO,CHPP,CWPP,RHO,ACO,RG1,S1,T1,TM1,RG2,S2,T2,TM2
 172  FORMAT(A1,3F6.1,4F6.2,2F6.1,F5.1,F6.1,F4.1,F6.2,
     1 F4.1,F4.2,F3.1,2(F5.2,F4.2,2E11.4))
      ENDFILE 7
C
C
      WRITE(2,600)
 600  FORMAT(/'+',80('*')//'+',2X,'PROGRAM PLUME  developed by :',
     1 '  Ashok S. Damle'/'+',2X,'Version 1.0',20X,'Research Triangle
     2 Institute'/'+',33X,'P.O.Box 12194'/'+',2X,'December
     3 1984',18X,'Research Triangle Park, NC 27709'//'+',20X, 'Copyright

     4 (C) 1984  Dr. Ashok S. Damle'/'+',20X,'All Commercial
     5 Rights Reserved'//'+', 80('*')//)
      WRITE(2,610) FILN
 610  FORMAT(/28X,'PLUME OPACITY MODEL PREDICTIONS'///
     1 5X,'DATA SET -',13A1/)
      RPO=DPO/2.0
      THETAO=DEGO*1.74533E-2
      CHPO=1.21942E-5*CHPP/(TPO+273.15)
      CWPO=1.21942E-1*CWPP/(TPO+273.15)
      CHA=1.21942E-5*CHAP/(TA+273.15)
      CWA=1.6045E-4*RH*(EXP(21.505-5464.5/(TA+273.15)))/(TA+2 73.15)
      T=0.0
      TPR=0.0
      ID=1
      N1=N+1
      NN=N+7
      M=0
      C1=T1/ALOG(S1)/2.5066283
      C2=T2/ALOG(S2)/2.5066283
      D1=2.0*(ALOG(S1))**2
      D2=2.0*(ALOG(S2))**2
C
C     Initialize all the plume variables
C
      DO 35 I=1,N
      A(I)=C1*EXP(-1.0*(ALOG(R(I)/RG1))**2/D1)*DLNR
  35  A(I)=A(I)+C2*EXP(-1.*(ALOG(R(I)/RG2))**2/D2)*DLNR
      A(N+1)=RPO
      A(N+2)=UO
      A(N+3)=THETAO
      A(N+4)=TPO
      A(N+5)=MWPO
      A(N+6)=CWPO
      A(N+7)=CHPO
      A(N+8)=0.0
      A(N+9)=0.0
      WRITE(2,900)
C
C     Print initial problem parameters
C
 900  FORMAT(/2X,'INITIAL BIMODAL SIZE DISTRIBUTION PARAMETERS -')
      WRITE(2,901) RG1,S1,T1
 901  FORMAT(2X,'GEOMETRIC MEAN RADIUS(UM)=',1PE9.2,1X,
     1 'STD.DEV.=',
     2 0PF4.1,1X,'NO.CONC.(#/CM3)=',1PE9.2)
      WRITE(2,901) RG2,S2,T2
      WRITE (2,902) UA,TA,RH,CHAP
 902  FORMAT(/2X,'AMBIENT PROPERTIES -'/2X,'WIND SPEED(M/SEC) = ',
     1 F6.2,3X,'AMBIENT TEMP.(C)=',F6.2/2X,'RELATIVE HUMIDITY (%) = ',
     2 F5.1,3X,'SULFURIC ACID CONC.(PPM)=',F5.2)
      WRITE(2,630)
 630  FORMAT(/2X,'PARTICLE RADII IN 25 SIZE CHANNELS(UM) -')
      WRITE(2,640) (R(I),I=1,N)
 640  FORMAT(5(2X,1PE10.3))
      WRITE(2,650)
 650  FORMAT(//)
      GO TO 20
C
C     Start looping with respect to time step
C
 2    IMAX=1
C
C     Determine maximum derivative function to set
C     proper time step
C
      B1=A(1)*V(1)
      DO 3 I=2,N
      B2=A(I)*V(I)
      IF(B2.GT.B1) IMAX=I
 3    IF(B2.GT.B1) B1=B2
      CALL DERV(A,F,R,V,ID,N,UA,CWA,CHA,MWA,TA,CPA,
     1 AL1,AL2,AL3,ACO)
      DEMAX=ABS(F(IMAX))/A(IMAX)
      DO 6 I=N1,NN
      IF(A(I).LE.1.0E-10) GOTO 6
      DE=ABS(F(I))/A(I)
      IF(DE.GT.DEMAX) DEMAX=DE
 6    CONTINUE
      DT=0.1/DEMAX
      IF(DTOLD.LT.DT) DT=DTOLD
C
C     Solve simultaneous ordinary differential
C     equations by 4th order Runge-Kutta algorithm
C
 10   CALL RUNGE(A,F,T,DT,IS,M,NT)
```

```
      DO 15 I=1,NT
 15   IF(A(I).LT.0.0) A(I)=0.0
      IF(IS.NE.1) GOTO 20
      CALL DERV(A,F,R,V,ID,N,UA,CWA,CHA,MWA,TA,CPA,
     1 AL1,AL2,AL3,ACO)
      GOTO 10
 20   IF((A(32)/CHPO).LE.0.001) GOTO 210
      IF(T.GE.TEND) GOTO 210
      IF(T.LT.TPR) GOTO 2
C
C     Print output results at selected time interval
C
      TPR=T+DTPR
 210  WRITE(2,740) T
 740  FORMAT(/2X,'TIME(SEC)=',1PE12.4)
      WRITE(2,601)
 601  FORMAT(/2X,'PLUME VARIABLES -'/
     1 2X,'NO. CONC. IN 25 SIZE CHANNELS (NO./CM3) -')
      WRITE(2,602) (A(I),I=1,N)
 602  FORMAT(5(2X,1PE9.2))
      WRITE(2,603) (A(I),I=N1,NT)
 603  FORMAT(/2X,'PLUME RADIUS(M)=',F6.2,2X,'PLUME VELOCITY (M/SEC) =',
     1 F6.2,2X,'PLUME ANGLE(RAD)=',F6.2/2X,'PLUME TEMP.(C)=',
     2 F6.2,3X,'MOL.WT. OF PLUME GAS=',F6.2/2X,'WATER VAPOR
     3 CONC. (GMOLE/LIT)=',1PE9.2/2X,'SULFURIC ACID VAPOR
     4 CONC. (GMOLE/LIT)=',1PE9.2/2X,'DISTANCE TRAVELLED BY
     5 PLUME : VERTICAL(M)=',0PF6.2,3X,
     6 'HORIZONTAL(M)=',0PF6.2)
      VF=3.141593*A(27)*A(26)**2
      WRITE(2,780) VF
 780  FORMAT(2X,'PLUME VOL. FLOW (M3/SEC)=',1PE9.2)
C
C
      TN=0.0
      TS=0.0
      PHI=0.0
      DO 30 I=1,N
      TN=TN+A(I)
      TS=TS+A(I)*R(I)*R(I)
 30   PHI=PHI+A(I)*V(I)
      TS=TS*12.56637
      PHIM=PHI*1.0E-3
      WRITE(2,760) TN,PHIM
 760  FORMAT(2X,'TOTAL NO. CONC.(NO./CM3)=',1PE9.2,2X,
     1 'TOTAL MASS CONC.(MG/M3)=',1PE9.2)
      WRITE(2,800) TS
 800  FORMAT(2X,'TOTAL SURFACE AREA TS (UM2/CM3)=',1PE9.2)
C
C     Calculate the opacity of the plume
C
      CALL OPACTY(GAMMA,R,A,DLNR,VISRED)
      VISRED=1.0-VISRED
      WRITE(2,790) GAMMA,VISRED
 790  FORMAT(2X,'EXTINCTION COEFF.(GAMMA)=',1PE12.4/2X,
     1'OPACITY (FRACTION)=',1PE12.4/)
C
C     Check for plume travel time, and
C     vapor species concentration
C
  200 IF((A(32)/CHPO).LE.0.002) GOTO 300
      IF(T.GE.TEND) GOTO 300
      GOTO 2
  300 WRITE(3,1005)
 1005 FORMAT(' ENTER NO. OF DATASETS REMAINING'/)
      READ(3,1006) NUM
 1006 FORMAT(I3)
      IF(NUM.NE.0) GOTO 173
      STOP
      END

C
C     SUBROUTINE DERFCN
C
C     This subroutine calculates the derivative function
C     for all the plume variables. This subroutine calls
C     subroutine GROWTH to obtain rate of vapor condensation
C     and corrosponding change in particlulate size
C     distribution and concentration. This subroutine
C     calculates the coagulation contributions to the
C     size spectrum depending upon coagulation indicator
C     ACO.
C
C
      SUBROUTINE DERV(A,F,R,V,ID,N,UA,CWA,CHA,WA,TA,CPA,
     1 AL1,AL2,AL3,ACO)
      DIMENSION A(34),F(34),R(25),V(25),BETA1(25,25)
      DIMENSION BETA2(25,25),CR(25),D(25),G(25),DE(25)
      DIMENSION CPA(25),GR(25),FKN(25)
      TP=A(29)
      IF(ID.EQ.0) GOTO 160
      XSAVE=0.9
      TPI=TP
      ID=0
  160 IF(TPI.LT.TP) GOTO 50
      TPI=TP-10.0
      TPA=TP+273.15
      VIS=1.45E-5*TPA**1.5/(TPA+118.0)
      FPA=21.4611*VIS*SQRT(TPA)
      DO 10 I=1,N
      FPBYR=FPA/R(I)
      FKN(I)=(1.0+FPBYR)/(1.+1.71*FPBYR+1.333*FPBYR**2)
   10 CONTINUE
      IF(ACO.NE.1.0) GOTO 50
      DO 11 I=1,N
      FPBYR=FPA/R(I)
      CR(I)=1.0+FPBYR*(1.257+0.4*EXP(-1.1/FPBYR))
      D(I)=7.321E-14*TP*CR(I)/R(I)/VIS
      B=1.38E-4*TP/V(I)
      G(I)=SQRT(B)*1.633
      FP=G(I)*D(I)/B*10000.0
      DE(I)=2.357E-5*((2.0*R(I)+FP)**3-(4*R(I)*R(I)+FP*FP)**
     1 1.5)/FP/R(I)
      DE(I)=DE(I)-2.8284E-4*R(I)
 11   CONTINUE
      DO 15 I=1,N
      DO 20 J=1,I
      IF(J.EQ.I) GOTO 30
      RM=(R(I)+R(J))/20000.0
      DM=(D(I)+D(J))/2.0
      GM=SQRT(G(I)*G(I)+G(J)*G(J))
      DEM=SQRT(DE(I)*DE(I)+DE(J)*DE(J))
      GOTO 40
 30   RM=R(I)/10000.0
      DM=D(I)
      GM=1.414*G(I)
      DEM=1.414*DE(I)
 40   BET=9.2E-17*TP/VIS/(RM/(RM+DEM/2.0)+4.0*DM/RM/GM)
      BETA2(I,J)=BET*(R(I)+R(J))*(CR(I)/R(I)+CR(J)/R(J))
      BETA1(I,J)=BETA2(I,J)*V(J)/V(I)
 20   BETA2(J,I)=BETA2(I,J)
 15   CONTINUE
 50   IF(ACO.NE.1.0) GOTO 52
      DO 60 I=1,N
      SUM1=0.0
      IF(I.EQ.1) GOTO 70
      L=I-1
      DO 80 J=1,L
 80   SUM1=SUM1+BETA1(I,J)*A(J)
      SUM1=SUM1*A(I)
 70   SUM2=0.0
      IF(I.EQ.N) GOTO 90
      DO 100 J=I,N
 100  SUM2=SUM2+BETA2(J,I)*A(J)
      SUM2=SUM2*A(I)
 90   CONTINUE
      IF(I.EQ.1) GOTO 65
      F(I)=SUM1-SUM2+BETA2(I-1,I-1)*A(I-1)**2/2.82840
      GOTO 60
 65   F(I)=-SUM2
 60   CONTINUE
      GOTO 62
 52   DO 54 I=1,N
 54   F(I)=0.0
 62   CALL GROWTH(A,R,V,FKN,GR,HLOSS,XSAVE,N)
      VE=AL1*(A(27)-UA*COS(A(28)))+AL2*UA*SIN(A(28))
      VE=VE+AL3*(TA-A(29))*A(26)/A(27)/(A(29)+273.15)
      C1=2.0*VE/A(26)
      C2=WA/A(30)
      C3=(A(29)+273.15)/(TA+273.15)
      DO 120 I=1,N
 120  F(I)=F(I)+GR(I)+C1*CPA(I)-C1*A(I)*(C2+C3-C2*C3)
      F(26)=VE*(C2+C3*(1.0-C2*UA*COS(A(28))/A(27)))
      F(26)=F(26)-4.9*SIN(A(28))*A(26)/A(27)*(C2*C3-1.)
      F(27)=C1*C2*C3*(UA*COS(A(28))-A(27))
      F(27)=F(27)+9.8*SIN(A(28))*(C2*C3-1.0)
      F(28)=9.8*COS(A(28))*(C2*C3-1.0)/A(27)
      F(28)=F(28)-C1*C2*C3*UA*SIN(A(28))/A(27)
      F(29)=C1*C2*C3*(TA-A(29))
      F(30)=C1*C3*(WA-A(30))
      F(31)=C1*CWA-C1*A(31)*(C2+C3-C2*C3)
      F(32)=C1*CHA-C1*A(32)*(C2+C3-C2*C3)-HLOSS
      F(33)=A(27)*SIN(A(28))
      F(34)=A(27)*COS(A(28))
      RETURN
      END

C
C     SUBROUTINE GROWTH
C
C     This subroutine calculates the rate of condensation
C     of vapors and corresponding change in particulate
C     concentration and size distribution for
C     Sulfuric acid - Water binary vapor system.
C     Both heterogeneous condensation and homogeneous
C     nucleation of vapors is considered, however, the
C     respective rates of condensation are assumed to
C     be controlled by sulfuric acid vapor only.
C
C
      SUBROUTINE GROWTH(A,R,V,FKN,GR,HLOSS,XSAVE,N)
      DIMENSION A(34),R(25),V(25),GR(25),G(25),FKN(25)
      TP=A(29)
      IF(A(32).LE.0.0) GOTO 250
      PWP=A(31)*(TP+273.15)*62.32473
      PHP=A(32)*(TP+273.15)*62.32473
      IF(TP.GT.155.0) GOTO 250
      IF(TP.LT.70.0) GOTO 10
      IF(TP.LT.125.0) GOTO 20
      AW=-4.155-0.0204*TP+5.61E-5*TP*TP
      BW=53.558-0.219*TP+9.95E-4*TP*TP
      CW=-64.16+0.397*TP-1.44E-3*TP*TP
      GOTO 30
 20   AW=-13.5+0.146*TP-4.0E-4*TP*TP
      BW=54.6-0.314*TP+9.99E-4*TP*TP
      CW=-50.9+0.245*TP-6.42E-4*TP*TP
      GOTO 30
 10   AW=-9.39+0.147*TP-5.07E-4*TP*TP
      BW=42.2-0.292*TP+1.21E-3*TP*TP
      CW=-44.5+0.27*TP-1.0E-3*TP*TP
 30   AH=-137.0+1.3*TP-4.07E-3*TP*TP
      BH=258.0-2.65*TP+8.73E-3*TP*TP
      CH=-132.0+1.48*TP-4.97E-3*TP*TP
      X1=XSAVE
      E1=PWP-EXP(AW+BW*X1+CW*X1*X1)
      RE1=ABS(E1/PWP)
      IF(RE1.LT.0.001) GOTO 50
      X2=X1-0.1
      E2=PWP-EXP(AW+BW*X2+CW*X2*X2)
      RE2=ABS(E2/PWP)
      IF(RE2.LT.0.001) GOTO 60
40    X=X1-E1*(X1-X2)/(E1-E2)
      IF(X.LT.0.45) X=0.45
      IF(X.GT.0.99) X=0.99
      E=PWP-EXP(AW+BW*X+CW*X*X)
      RE=ABS(E/PWP)
      IF(RE.LT.0.001) GOTO 70
      X1=X2
      E1=E2
      X2=X
      E2=E
      GOTO 40
50    X=X1
      GOTO 70
60    X=X2
70    PHE=EXP(AH+BH*X+CH*X*X)
      IF(TP.LT.65.0) PHE=0.0
      XSAVE=X
      DELP=PHP-PHE
      IF(DELP.GT.0.0) GOTO 80
250   DO 75 I=1,N
75    GR(I)=0.0
      HLOSS=0.0
      GOTO 500
80    XM=18.0*X/(98.0-80.0*X)
      CON1=4.856447E14/(TP+273.15)
      VH=1.597E-10
      VW=2.989E-11
      CON2=CON1*(VH+(1.0-XM)*VW/XM)/1.8284
      SUM=0.0
      L=N-1
      DO 90 I=1,L
      DELP=PHP-PHE*EXP(0.964/R(I)/(TP+273.15))
      IF(DELP.LE.0.0) GOTO 95
      B=R(I)*FKN(I)*A(I)*DELP
      G(I)=B/V(I)
      GOTO 90
95    B=0.0
      G(I)=0.0
90    SUM=SUM+B
      GR(1)=-CON2*G(1)
      GR(N)=CON2*G(N-1)
      DO 100 I=2,L
100   GR(I)=CON2*(G(I-1)-G(I))
      HLOSS=CON1*SUM/6.023E20
      IF(PHE.LE.0.0) GOTO 500
      SUP=PHP/PHE
      IF(SUP.LT.2.5) GOTO 500
      ASUP=ALOG(SUP)
      HOCON=1.426E31*A(32)**2*EXP(-6.3E9/(TP+273.15)**3/
     1 ASUP**2)
```

```
      RCRIT=0.644/(TP+273.15)/ASUP
      GR(1)=GR(1)+HOCON*(RCRIT/R(1))**3*(1.0+(1.0-XM)*VW/VH/
     1 XM)
      HLOSS=HLOSS+5.10815E-11*HOCON*RCRIT**3/(TP+273.15)
  500 RETURN
      END

      SUBROUTINE RUNGE(C,F,T,DT,IS,M,N)
      DIMENSION C(34),F(34),FF(34),SAVEC(34)
      M=M+1
      GO TO (110,130,150,170),M
  110 DO 120 I=1,N
      SAVEC(I)=C(I)
      FF(I)=F(I)
  120 C(I)=SAVEC(I)+0.5*DT*F(I)
      T=T+0.5*DT
      IS=1
      GOTO 190
  130 DO 140 I=1,N
      FF(I)=FF(I)+2.0*F(I)
  140 C(I)=SAVEC(I)+0.5*DT*F(I)
      IS=1
      GOTO 190
  150 DO 160 I=1,N
      FF(I)=FF(I)+2.0*F(I)
  160 C(I)=SAVEC(I)+DT*F(I)
      T=T+0.5*DT
      IS=1
      GOTO 190
  170 DO 180 I=1,N
  180 C(I)=SAVEC(I)+(FF(I)+F(I))*DT/6.0
      M=0
      IS=2
  190 CONTINUE
      RETURN
      END

      SUBROUTINE OPACTY(GAMMA,R,A,DLNR,VISRED)
      DIMENSION R(25),A(34),Q(25)
      DATA Q/0.0027,0.0045,0.0076,0.0123,0.0205,0.0355,
     1 0.06032,0.12211,0.3,0.75405,1.83651,3.10027,
     2 3.83479,3.09179,2.20888,2.60279,2.34689,2.27771,
     3 2.20221,6*2.0/
      GAMMA=0.0
      DO 10 I=1,25
   10 GAMMA=GAMMA+Q(I)*A(I)*R(I)*R(I)
      GAMMA=GAMMA*DLNR*3.14159E-8
      VISRED=EXP(-GAMMA*A(26)*200.0)
      RETURN
      END

      SUBROUTINE CLS
      LOGICAL HOME,CLEAR
      HOME=28
      CLEAR=31
      WRITE(3,10)HOME,CLEAR
10    FORMAT('+',2A1)
      RETURN
      END
```

```
C
C **********************************************
C
C     COPYRIGHT (C) 1984  DR. ASHOK S. DAMLE
C                         MR. ANDREW S. VINER
C
C     ALL COMMERCIAL RIGHTS RESERVED
C
C **********************************************
C
C PROGRAM PLUMEIN developed by : Dr. Ashok S. Damle
C                                Mr. Andrew S. Viner
C Version 1.0                    Research Triangle Institute
C                                P.O.Box 12194
C December 1984                  Res. Tri. Park,  NC 27709
C
C                                Phone: (919) 541-5829
C
C **********************************************
C
C     PROGRAM PLUMEIN TO PREPARE AN INPUT DATA FILE
C
      PROGRAM PLUMEIN
      DIMENSION LABEL(40),HEAD(40)
      REAL MWPO
      INTEGER HEAD,POINTR,ELMNTS,ROW,COLUMN
      LOGICAL HOME,SFLAG(10),HCOPY,FILN(13),CONTRL
      COMMON /INITAL/UA,TA,RH,CHAP,AL1,AL2,AL3,DPO,
    1 UO,DEGO,TPO,MWPO,CHPP,CWPP,RHO,ACO,RG1,S1,T1,
    2 TM1,RG2,S2,T2,TM2
      DATA HEAD,FILN/40*32,13*32/
      CALL DFAULT
      JPOINT=1
      CONV1=2.387E2
      HOME=X'1C'
1     CALL CLS
      DO 4 I=1,10
      SFLAG(I)=.TRUE.
4     CONTINUE
      WRITE(3,100)HOME
100   FORMAT(
    1 '+',1A1,63('*')
    2 /'+',22X,'RTI Plume model'
    3 /'+',63('*')
    5 /'+',20X,'Input Ambient parameters'
    5 /'+',20X,'Input Plume parameters'
    6 /'+',20X,'Save Input Parameters in a File'
    B /'+',20X,'Read Input Parameters from a File'
    D /'+',20X,'Exit'
    E ///'+',20X,'Press H for HELP')
      POINTR=JPOINT
      ELMNTS=5
2     CALL MENU(ELMNTS,POINTR,SFLAG,ROW,COLUMN,LETTER)
```

```
      IF(LETTER.EQ.72)CALL HELP
      IF(LETTER.EQ.72)GOTO 1
      IF(LETTER.NE.13)GOTO 2
      DO 3 I=1,10
3     SFLAG(I)=.FALSE.
      JPOINT=POINTR
      GOTO(20,30,60,60,92),POINTR
20    POINTR=1
      ELMNTS=7
11    CALL CLS
      WRITE(3,110)HOME
110   FORMAT('+',1A1,63('#')
     2 /'+Ambient Parameters',34X,'Page 1 of 1'
     3 /'+',63('#'))
      WRITE(3,111)UA,TA,RH,CHAP,AL1,AL2,AL3
111   FORMAT(
     6 '+Wind Speed',36X,F6.1,'   m/sec'
     7 /'+Ambient temp. at stack height',17X,F6.1,' degrees C'
     8 /'+Percent Relative Humidity',21X,F6.1,'  %'
     8 /'+Amb. sulfuric acid vapor conc.',17X,F6.2,' ppm'
     1 /'+Entrainment Parameter Alpha1',19X,F6.2
     2 /'+                      Alpha2',19X,F6.2
     3 /'+                      Alpha3',19X,F6.2
     D /
     E /'+Press H for HELP')
12    CALL MENU(ELMNTS,POINTR,SFLAG,ROW,COLUMN,LETTER)
      IF(LETTER.EQ.82)GOTO 1
      IF(LETTER.EQ.72)CALL HELP1
      IF(LETTER.EQ.72)GOTO 11
      IF(.NOT.(((LETTER.GE.48).AND.(LETTER.LE.57))
     1 .OR.(LETTER.EQ.45)
     2 .OR.(LETTER.EQ.46)))GOTO 12
      Z=VALUE(ROW,COLUMN,LETTER)
      IF(POINTR.EQ.1)UA=Z
      IF(POINTR.EQ.2)TA=Z
      IF(POINTR.EQ.3)RH=Z
      IF(POINTR.EQ.4)CHAP=Z
      IF(POINTR.EQ.5)AL1=Z
      IF(POINTR.EQ.6)AL2=Z
      IF(POINTR.EQ.7)AL3=Z
      GOTO 11
30    POINTR=1
      ELMNTS=9
31    CALL CLS
      WRITE(3,130)HOME
130   FORMAT('+',1A1,63('#')
     2 /'+Plume Parameters',36X,'Page 1 of 2'
     3 /'+',63('#'))
      WRITE(3,131)DPO,UO,DEGO,TPO,MWPO,CHPP,CWPP,RHO,ACO
131   FORMAT(
     5 '+Stack Exit Diameter',27X,F6.1,'    m'
     6 /'+Exit velocity from stack',22X,F6.1,' m/sec'
     7 /'+Plume inclination at stack exit',16X,F5.1,' degrees'
     8 /'+Plume temp. at stack exit',21X,F6.1,' degrees C'
     9  /'+Molecular weight of exit gas',20X,F4.1
     A  /'+Sulfuric acid vapor concentration',14X,F6.2,' ppm'
     B  /'+Water vapor concentration',23X,F4.1,' % volume'
     E  /'+Particle density',32X,F5.2,' g/cc'
     1  /'+Indicator to include Coagulation in'
     2  /'+  calculations, 1=YES, 2=NO',22X,F3.1
     C  /
     D  /'+Use > to go to Page 2.  Press H for HELP')
 32     CALL MENU(ELMNTS,POINTR,SFLAG,ROW,COLUMN,LETTER)
        IF(LETTER.EQ.82)GOTO 1
        IF(LETTER.EQ.62)GOTO 33
        IF(LETTER.EQ.72)CALL HELP2
        IF(LETTER.EQ.72)GOTO 31
        IF(.NOT.(((LETTER.GE.48).AND.(LETTER.LE.57))
     1  .OR.(LETTER.EQ.45)
     2  .OR.(LETTER.EQ.46)))GOTO 32
        Z=VALUE(ROW,COLUMN,LETTER)
        IF(POINTR.EQ.1)DPO=Z
        IF(POINTR.EQ.2)UO=Z
        IF(POINTR.EQ.3)DEGO=Z
        IF(POINTR.EQ.4)TPO=Z
        IF(POINTR.EQ.5)MWPO=Z
        IF(POINTR.EQ.6)CHPP=Z
        IF(POINTR.EQ.7)CWPP=Z
        IF(POINTR.EQ.8)RHO=Z
        IF(POINTR.EQ.9)ACO=Z
        GOTO 31
 33     POINTR=1
        ELMNTS=8
 34     CALL CLS
        WRITE(3,135)HOME
 135    FORMAT('+',1A1,63('#')
     2  /'+plume parameters',36X,'page 2 of 2'
     3  /'+',63('#'))
        WRITE(3,136)RG1,S1,T1,TM1,RG2,S2,T2,TM2
 136    FORMAT(
     5  '+Mode 1 - Geometric Mean Radius',14X,F5.2,' microns'
     6  /'+          Geometric std. deviation',12X,F4.2
     7  /'+          Concentration (number)',13X,1PE11.4,'
no./cc'
     8  /'+          Concentration (mass)',15X,1PE11.4,' mg/m3'
     9  /'+Mode 2 - Geometric Mean Radius',14X,0PF5.2,'
microns'
     A  /'+          Geometric std. deviation',12X,F4.2
     B  /'+          Concentration (number)',13X,1PE11.4,'
no./cc'
     C  /'+          Concentration (mass)',15X,1PE11.4,' mg/m3'
     E  /
     F  /'+Use < to go to Page 1. Press H for HELP')
 35     CALL MENU(ELMNTS,POINTR,SFLAG,ROW,COLUMN,LETTER)
        IF(LETTER.EQ.82)GOTO 1
        IF(LETTER.EQ.60)GOTO 30
        IF(LETTER.EQ.72)CALL HELP2
        IF(LETTER.EQ.72)GOTO 34
        IF(.NOT.(((LETTER.GE.48).AND.(LETTER.LE.57))
     1  .OR.(LETTER.EQ.45)
     2  .OR.(LETTER.EQ.46)))GOTO 35
        Z=VALUE(ROW,COLUMN,LETTER)
        IF(POINTR.EQ.1)RG1=Z
        IF(POINTR.EQ.2)S1=Z
        IF(POINTR.EQ.3)GOTO 21
        IF(POINTR.EQ.4)GOTO 22
        IF(POINTR.EQ.5)RG2=Z
        IF(POINTR.EQ.6)S2=Z
        IF(POINTR.EQ.7)GOTO 23
        IF(POINTR.EQ.8)GOTO 24
        GOTO 34
 21     T1=Z
        IF(S1.LE.0)GOTO 34
        TM1=EXP(4.5*ALOG(S1)**2)*T1*RHO*RG1**3/CONV1
        GOTO 34
 22     TM1=Z
        IF(S1.LE.0)GOTO 34
        IF((RHO*RG1).EQ.0)GOTO 34
        T1=CONV1*TM1/(EXP(4.5*ALOG(S1)**2)*RHO*RG1**3)
        GOTO 34
 23     T2=Z
        IF(S2.LE.0)GOTO 34
        TM2=EXP(4.5*ALOG(S2)**2)*T2*RHO*RG2**3/CONV1
        GOTO 34
 24     TM2=Z
        IF(S2.LE.0)GOTO 34
        IF((RHO*RG2).EQ.0)GOTO 34
        T2=CONV1*TM2/(EXP(4.5*ALOG(S2)**2)*RHO*RG2**3)
        GOTO 34
 60     CALL CLS
        WRITE(3,160)
 160    FORMAT(//'+',10X,'Enter a Filename (e.g. TEST1/DAT)'
     1  //'+',14X,'or press <ENTER> to quit')
        ROW=7
        COLUMN=14
        LETTER=95
        CALL DBOUNC
        CALL BLINKR(ROW,COLUMN,LETTER)
        IF(LETTER.LT.65.OR.LETTER.GT.90)GOTO 1
        CALL STRNG(LABEL,ROW,COLUMN,LETTER,NCHAR)
        IF(LABEL(1).EQ.32)GOTO 1
        DO 61 I=1,12
        FILN(I)=LABEL(I)
 61     LABEL(I)=32
        CALL OPEN(7,FILN,0)
        IF(POINTR.EQ.4)GOTO 70
        WRITE(7,161,REC=1)UA,TA,RH,CHAP,AL1,AL2,AL3,DPO,UO,DE
GO,TPO,
     1  MWPO,CHPP,CWPP,RHO,ACO,RG1,S1,T1,TM1,RG2,S2,T2,TM2
 161    FORMAT(' ',3F6.1,4F6.2,2F6.1,F5.1,F6.1,F4.1,F6.2,
     1  F4.1,F4.2,F3.1,2(0PF5.2,F4.2,1PE11.4,1PE11.4))
        ENDFILE 7
```

```
      CALL PRINTI(FILN)
      GOTO 1
 70   READ(7,170,REC=1)CONTRL,UA,TA,RH,CHAP,AL1,AL2,AL3,
     1 DPO,UO,DEGO,TPO,MWPO,CHPP,CWPP,RHO,ACO,RG1,S1,
     2 T1,TM1,RG2,S2,T2,TM2
 170  FORMAT(A1,3F6.1,4F6.2,2F6.1,F5.1,F6.1,F4.1,F6.2,F4.1,
     1 F4.2,F3.1,2(F5.2,F4.2,2E11.4))
      ENDFILE 7
      CALL PRINTI(FILN)
      GOTO 1
 92   CONTINUE
      END
      SUBROUTINE PRINTI(FILN)
      REAL MWPO
      LOGICAL FILN(13),DATE(8),TIME(8),FF
      COMMON /INITAL/UA,TA,RH,CHAP,AL1,AL2,AL3,DPO,
     1 UO,DEGO,TPO,MWPO,CHPP,CWPP,RHO,ACO,RG1,S1,T1,
     2 TM1,RG2,S2,T2,TM2
      FF=12
      CALL CLS
      WRITE(3,100)
 100  FORMAT('+Would you like a hardcopy printout ?')
 1    I=2
      J=35
      L=63
      CALL DBOUNC
      CALL BLINKR(I,J,L)
      IF(L.EQ.78)RETURN
      IF(L.EQ.89)GOTO 2
      WRITE(3,101)
 101  FORMAT(/'+Please answer Y or N')
      GOTO 1
 2    CALL DBOUNC
      CALL DATIME(DATE,TIME)
      IF((PEEK(14312).AND.240).EQ.48)GOTO 3
      WRITE(3,102)
 102  FORMAT('+Press <ENTER> when Line Printer is Ready')
      I=INKEY(J)
 3    WRITE(2,200)FILN,DATE,TIME
 200  FORMAT('+',80('*')//'+Summary of Input Data for
     1Plume Model',10X,'Filename: '13A1//'+',80('*')
     2 //'+',60X,'Date ',8A1/'+',60X,'Time ',8A1)
      WRITE(2,201)UA,TA,RH,CHAP,AL1,AL2,AL3
 201  FORMAT(//'+Ambient Variables'/
     1 /'+',5X,'Wind Speed',42X,F6.1,'  m/sec'
     2 /'+',5X,'Ambient temperature',33X,F6.1,'  C'
C The sole purpose of this line is to lenghten the file
     3 /'+',5X,'Percent Relative Humidity',27X,F6.1,'  %'
     4 /'+',5X,'Ambient sulfuric acid vapor concentration',
     5 12X,F6.2,' ppm'
     6 /'+',5X,'Entrainment Parameters,  Alpha1',22X,F6.2
     7 /'+',30X,'Alpha2',22X,F6.2
     8 /'+',30X,'Alpha3',22X,F6.2)
      WRITE(2,208)DPO,UO,DEGO,TPO,MWPO,CHPP,CWPP,RHO
```

```
208     FORMAT(//'+Plume Variables'
     1  //'+',5X,'Stack Exit Diameter',33X,F6.1,'  m'
     2  /'+',5X,'Exit Velocity at stack',30X,F6.1,'  m/sec'
     3  /'+',5X,'Plume inclination at stack exit',22X,F5.1,'
degrees'
     4  /'+',5X,'Plume temperature at stack exit',21X,F6.1,'
degrees C'
     5  /'+',5X,'Molecular weight of exit gas',26X,F4.1
     6  /'+',5X,'Sulfuric acid vapor concentration',20X,F6.2,'
ppm'
     7  /'+',5X,'Water vapor concentration',29X,F4.1,'  %
volume'
     8  /'+',5X,'Particle density',39X,F4.2,' gm/cm3')
        WRITE(2,216)RG1,S1,T1,TM1,RG2,S2,T2,TM2,ACO,FF
 216    FORMAT(//'+Initial particulate size distribution and
     1 concentration'/
     2  /'+',5X,'Mode 1 - Geometric Mean Radius',24X,F5.2,'
microns'
     2  /'+',5X,'         Geometric Std. Deviation',22X,F4.2,'
'
     3 /'+',5X,'         Concentration (number)',23X,1PE11.4,'
/cm3'
     4  /'+',5X,'         Concentration (mass)',25X,E11.4,'
mg/m3'/
     5  /'+',5X,'Mode 2 - Geometric Mean Radius',24X,0PF5.2,'
microns'
     6  /'+',5X,'         Geometric Std. Deviation',22X,F4.2,'
'
     7 /'+',5X,'         Concentration (number)',23X,1PE11.4,'
/cm3'
     8 /'+',5X,'         Concentration (mass)',25X,E11.4,'
mg/m3',
     9  ///'+',5X,'Indicator to include Coagulation in'/
     1  '+',5X,'  calculations,  1=YES, 2=NO',27X,0PF3.1,A1)
        RETURN
        END
        SUBROUTINE STRNG(LABEL,ROW,COLUMN,LETTER,NCHAR)
        DIMENSION LABEL(40)
        INTEGER ROW,COLUMN,ADDRSS
        DO 1 I=1,40
 1      LABEL(I)=32
        LABEL(1)=LETTER
        ADDRSS=15360+(ROW-1)*64+COLUMN
        CALL POKE(ADDRSS,LETTER)
        NCHAR=1
        COLUMN=COLUMN+1
        ADDRSS=ADDRSS+1
 8      LETTER=95
        CALL DBOUNC
        CALL BLINKR(ROW,COLUMN,LETTER)
        IF(LETTER .EQ. 8)GOTO 10
        IF(LETTER .EQ. 13)RETURN
        NCHAR=NCHAR+1
        LABEL(NCHAR)=LETTER
```

```
         CALL POKE(ADDRSS,LETTER)
         ADDRSS=ADDRSS+1
         COLUMN=COLUMN+1
         IF(NCHAR .EQ. 40)RETURN
         GOTO 8
10       IF(NCHAR .EQ. 0)GOTO 8
         LABEL(NCHAR)=32
         NCHAR=NCHAR-1
         ADDRSS=ADDRSS-1
         COLUMN=COLUMN-1
         GOTO 8
         END

        SUBROUTINE DATIME(DATE,TIME)
        LOGICAL DATE(8),TIME(8),TEST,D(2)
        TEST=PEEK(X'421A')
        IF(TEST.NE.0)GOTO 100
        CALL CLS
1       WRITE(3,10)
10      FORMAT(/////'+DATE (MM/DD/YY) ')
        READ(3,20)DATE
20      FORMAT(8A1)
        WRITE(3,30)
30      FORMAT('+TIME (HH:MM:SS) ')
        READ(3,20)TIME
        DO 50 I=1,7,3
        D(1)=DATE(I)
        D(2)=DATE(I+1)
        DECODE(D,40)TEST
        K=1+I/3
        CALL POKE(X'421D'-K,TEST)
50      CONTINUE
        IF(TEST.EQ.0)WRITE(3,15)
15      FORMAT('+TRY AGAIN')
        IF(TEST.EQ.0)GOTO 1
        DO 60 I=1,7,3
        D(1)=TIME(I)
        D(2)=TIME(I+1)
        DECODE(D,40)TEST
        K=1+I/3
        CALL POKE(X'421A'-K,TEST)
60      CONTINUE
        IF(TEST.GT.60)WRITE(3,15)
        IF(TEST.GT.60)GOTO 1
40      FORMAT(I2)
        RETURN
100     DO 110 I=1,3
        TEST=PEEK(X'4219'+I)
        K=8-3*(I-1)
        DATE(K)=48+TEST-10*(TEST/10)
110     DATE(K-1)=48+TEST/10
        DATE(3)='/'
        DATE(6)='/'
120     FORMAT(2A1)
```

```
      DO 130 I=1,3
      TEST=PEEK(X'4216'+I)
      K=8-3*(I-1)
      TIME(K)=48+TEST-10*(TEST/10)
130   TIME(K-1)=48+TEST/10
      TIME(3)=':'
      TIME(6)=':'
      RETURN
      END
      SUBROUTINE CLS
      LOGICAL HOME,CLEAR
      HOME=28
      CLEAR=31
      WRITE(3,10)HOME,CLEAR
10    FORMAT('+',2A1)
      RETURN
      END

       SUBROUTINE DFAULT
       REAL MWPO
       COMMON /INITAL/UA,TA,RH,CHAP,AL1,AL2,AL3,DPO,
     1 U0,DEGO,TPO,MWPO,CHPP,CWPP,RHO,ACO,RG1,S1,T1,
     2 TM1,RG2,S2,T2,TM2
       UA=5.0
       TA=20.0
       RH=50.0
       CHAP=00.0
       AL1=0.11
       AL2=0.6
       AL3=0.0
       DPO=4.0
       U0=20.0
       DEGO=90.0
       TPO=200.0
       MWPO=29.00
       CHPP=10.0
       CWPP=5.0
       RHO=1.0
       ACO=2.0
       RG1=0.07
       S1=1.3
       T1=1.0E6
       TM1=0.0
       RG2=1.0
       S2=1.8
       T2=3.0E3
       TM2=0.0
       RETURN
       END

       SUBROUTINE DBOUNC
C      Delay loop to prevent key bounce
       L=0
       DO 1 I=1,4000
```

```
 1      L=L+1
        RETURN
        END
C
        SUBROUTINE BLINKR(ROW,COLUMN,FLASHR)
        INTEGER ROW,COLUMN,FLASHR,ADDRSS,INKEY
        CALL KBINIT
        ADDRSS=15360+(ROW-1)*64+COLUMN
        CALL POKE(ADDRSS,128)
        KOUNT=1
 2      LETTER=INKEY(0)
        IF(LETTER)6,3,6
 3      LETTER=MOD(KOUNT,20)
        IF(LETTER)5,4,5
 4      K=PEEK(ADDRSS)
        IF(K .AND. FLASHR)CALL POKE(ADDRSS,128)
        IF(K .AND. -128)CALL POKE(ADDRSS,FLASHR)
 5      KOUNT=KOUNT+1
        IF(KOUNT .GT. 1000)KOUNT=1
        GOTO 2
 6      CONTINUE
        FLASHR=LETTER
        CALL POKE(ADDRSS,128)
        RETURN
        END
C
        INTEGER FUNCTION INKEY(NSCAN)
C
        DIMENSION KYBRD(8,7,2)
        INTEGER ISHIFT,ADDRSS,START,ROW,COLUMN,SHIFT
C
        COMMON /ASCKEY/KYBRD
        INKEY=0
        START=X'3800'
        COLUMN=0
        ROW=0
 10     SHIFT=1
C
        DO 1 I=1,7
                ADDRSS=START+2**(I-1)
                INPUT=PEEK(ADDRSS)
                IF(PEEK(X'3880') .or. 0)SHIFT=2
                IF(INPUT .or. 0)COLUMN=ISHIFT(INPUT)
                ROW=I
                IF(INPUT .or. 0)goto 2
 1      CONTINUE
C
 2      IF((NSCAN .EQ. 0) .AND. (COLUMN .EQ. 0))RETURN
        IF(COLUMN .EQ. 0)GOTO 10
        INKEY=KYBRD(COLUMN,ROW,SHIFT)
        RETURN
        END
C
        INTEGER FUNCTION ISHIFT(I)
```

```
      L=.FALSE.
      L=L .OR. 1
      DO 1 J=1,7
      IF(I .AND. L)ISHIFT=J
 1    L=L*2
      IF(I .AND. -128)ISHIFT=8
      RETURN
      END
C
      SUBROUTINE MENU(ELMNTS,POINTR,SFLAG,ROW,COLUMN,
     1 LETTER)
      INTEGER ELMNTS,ROW,COLUMN,POINTR,SFLAG
      LOGICAL SFLAG(10)
 1    KDASH=15592+64*(POINTR-1)
      IF(SFLAG(POINTR))KDASH=KDASH-25
      KARROW=KDASH+1
      CALL POKE(KDASH,45)
      CALL POKE(KARROW,62)
      CALL DBOUNC
      ROW=POINTR+3
      COLUMN=41
      IF(SFLAG(POINTR))COLUMN=16
      LETTER=62
      CALL BLINKR(ROW,COLUMN,LETTER)
      CALL POKE(KDASH,128)
      IF((LETTER .NE. 10) .AND. (LETTER .NE. 91))RETURN
      CALL POKE(KARROW,128)
      I=POINTR
      IF((LETTER .EQ. 10) .AND. (I .LT. ELMNTS))POINTR=I+1
      IF((LETTER .EQ. 10) .AND. (I .EQ. ELMNTS))POINTR=1
      IF((LETTER .EQ. 91) .AND. (I .EQ. 1))POINTR=ELMNTS
      IF((LETTER .EQ. 91) .AND. (I .GT. 1))POINTR=I-1
      GOTO 1
      END
      FUNCTION VALUE(ROW,COLUMN,LETTER)
      DIMENSION NUMINP(11)
      INTEGER ROW,COLUMN,ADDRSS,FLAG
      DO 3 I=1,11
 3    NUMINP(I)=32
      VALUE=0.0
      NUMINP(1)=LETTER
      ADDRSS=15360+(ROW-1)*64+COLUMN
      CALL POKE(ADDRSS,LETTER)
      NCHAR=1
 1    FLAG=0
      CALL DBOUNC
      COLUMN=COLUMN+1
      ADDRSS=ADDRSS+1
 8    LETTER=95
      CALL BLINKR(ROW,COLUMN,LETTER)
      IF(LETTER.EQ.8)GOTO 10
      IF(((LETTER.GE.48).AND.(LETTER.LE.57))
     1.OR.(LETTER.EQ.45)
     2.OR.(LETTER.EQ.46)
     3.OR.(LETTER.EQ.69))FLAG=1
      IF(LETTER.EQ.13)GOTO 2
      IF(FLAG.EQ.0)GOTO 8
      NCHAR=NCHAR+1
      NUMINP(NCHAR)=LETTER
      CALL POKE(ADDRSS,LETTER)
      IF(NCHAR.EQ.11)GOTO 2
      GOTO 1
10    IF(NCHAR.EQ.0)GOTO 8
      NUMINP(NCHAR)=48
      NCHAR=NCHAR-1
      ADDRSS=ADDRSS-2
      COLUMN=COLUMN-2
      GOTO 1
2     IF(NCHAR.LT.11)NUMINP(NCHAR+1)=46
      DECODE(NUMINP,20)VALUE
20    FORMAT(E11.4)
      RETURN
      END
      SUBROUTINE KBINIT
      DIMENSION KYBRD(8,7,2)
      COMMON /ASCKEY/KYBRD
      LETTER=63
      DO 1 I=1,2
      DO 1 J=1,4
      DO 1 K=1,8
1     KYBRD(K,J,I)=LETTER+K+(J-1)*8+(I-1)*32
      LETTER=47
      DO 2 I=1,2
      DO 2 J=5,6
      DO 2 K=1,8
2     KYBRD(K,J,I)=LETTER+K+(J-5)*8-(I-1)*16
      DO 3 K=5,8
      KYCHNG=KYBRD(K,6,1)
      KYBRD(K,6,1)=KYBRD(K,6,2)
3     KYBRD(K,6,2)=KYCHNG
      KYBRD(1,7,1)=13
      KYBRD(2,7,1)=31
      KYBRD(3,7,1)=1
      KYBRD(4,7,1)=91
      KYBRD(5,7,1)=10
      KYBRD(6,7,1)=8
      KYBRD(7,7,1)=9
      KYBRD(8,7,1)=32
      KYBRD(1,7,2)=13
      KYBRD(2,7,2)=31
      KYBRD(3,7,2)=1
      KYBRD(4,7,2)=27
      KYBRD(5,7,2)=26
      KYBRD(6,7,2)=24
      KYBRD(7,7,2)=25
      KYBRD(8,7,2)=32
      RETURN
      END
```

```
        SUBROUTINE HELP
        CALL CLS
        WRITE(3,10)
10      FORMAT(
     1  '+     The previous page listed 5 commands available to
you :'
     2  /'+2 data inputs; store input parameters in a disk file;
'
     3  /'+retrieve input parameters from a disk file; and
Exit.'
     4  /'+To execute any of these commands move the pointer (
) to'
     5  /'+the proper row using the up and down arrow keys on
the left'
     6  /'+end of the keyboard and select the subroutine by
pressing the
     6  '/'+<ENTER> key.  For example, to enter parameters
describing'
     7  /'+ambient conditions near the stack, move the pointer
to the '
     7  /'+row labelled "Input Ambient parameters" and press the
<ENTER>
     8  '/'+key.  A list of variables will be displayed on the
screen
     9  '/'+and instructions will be given on how to enter
data.'
     B  /
     C  /
     D  /'+   Press any key to continue')
        CALL POKE(15667,45)
        CALL POKE(15668,62)
        J=1
        I=INKEY(J)
        CALL CLS
        WRITE(3,20)
20      FORMAT(
     1  /'+     When the program is first started, the
parameters are'
     2  /'+initialized with the values which are typical of
plumes.'
     3  /'+You can change any or all of the values in the
program.'
     3  /'+When all of the data has been entered it must be
saved in a'
     4  /'+data file.  Move the pointer to the row labelled
"Save Input'
     5  /'+Parameters in a file" and press <ENTER>.  Enter a
filename'
     6  /'+as requested.  The program will write your input
values to'
     7  /'+disk and ask you if you want a hardcopy of the
parameter'
     8  /'+values.  Control will be returned to the main menu
```

```
where you'
     9  /'+can enter new values for another data file.  When you
are'
     A  /'+finished entering data, use the "Exit" option to end
the '
     B  /'+program.'
     F  /'+        Press any key to continue')
        I=INKEY(J)
        RETURN
        END
        SUBROUTINE HELP1
        CALL CLS
        WRITE(3,10)
10      FORMAT(
     1  '+     This is a data input subroutine.  Parameters
whose'
     2  /'+values can be input are listed in the column on the
left of'
     3  /'+the previous page.  The current value for each
parameter is'
     4  /'+shown in the right hand column.  To change a value,
move the'
     5  /'+pointer to the proper row, type in the new value, and
press'
     6  /'+<ENTER>.  The new value will be stored in the program
and'
     7  /'+the data page will be rewritten showing the new
value.  You'
     8  /'+can change the values any number of times.  When you
are'
     9  /'+satisfied with all of the inputs press R to return to
the '
     A  /'+ main menu.'
     B  /
     C  /
     E  /'+   Press any key to continue')
        J=1
        I=INKEY(J)
        RETURN
        END
        SUBROUTINE HELP2
        CALL CLS
        WRITE(3,10)
10      FORMAT(
     1  '+     The data input for this subroutine is the same as
for'
     2  /'+the previous input subroutines except that now there
are two'
     3  /'+pages of data.  To get to the other page press < or
>.'
     4  /'+Remember that these symbols require use of the SHIFT
key.'
     5  /
     6  /
```

```
7  /'+   Press any key to continue')
   J=1
   I=INKEY(J)
   RETURN
   END
```

Page 1

```
      SUBROUTINE DATIME(DATE,TIME)
      LOGICAL DATE(8),TIME(8),TEST,D(2)
      TEST=PEEK(X'421A')
      IF(TEST.NE.0)GOTO 100
      CALL CLS
1     WRITE(3,10)
10    FORMAT(/////'+DATE (MM/DD/YY) ')
      READ(3,20)DATE
20    FORMAT(8A1)
      WRITE(3,30)
30    FORMAT('+TIME (HH:MM:SS) ')
      READ(3,20)TIME
      DO 50 I=1,7,3
      D(1)=DATE(I)
      D(2)=DATE(I+1)
      DECODE(D,40)TEST
      K=1+I/3
      CALL POKE(X'421D'-K,TEST)
50    CONTINUE
      IF(TEST.EQ.0)WRITE(3,15)
15    FORMAT('+TRY AGAIN')
      IF(TEST.EQ.0)GOTO 1
      DO 60 I=1,7,3
      D(1)=TIME(I)
      D(2)=TIME(I+1)
      DECODE(D,40)TEST
      K=1+I/3
      CALL POKE(X'421A'-K,TEST)
60    CONTINUE
      IF(TEST.GT.60)WRITE(3,15)
      IF(TEST.GT.60)GOTO 1
40    FORMAT(I2)
      RETURN
100   DO 110 I=1,3
      TEST=PEEK(X'4219'+I)
      K=8-3*(I-1)
      DATE(K)=48+TEST-10*(TEST/10)
110   DATE(K-1)=48+TEST/10
      DATE(3)='/'
      DATE(6)='/'
120   FORMAT(2A1)
      DO 130 I=1,3
      TEST=PEEK(X'4216'+I)
      K=8-3*(I-1)
      TIME(K)=48+TEST-10*(TEST/10)
130   TIME(K-1)=48+TEST/10
      TIME(3)=':'
      TIME(6)=':'
      RETURN
      END
      SUBROUTINE CLS
      LOGICAL HOME,CLEAR
      HOME=28
      CLEAR=31
      WRITE(3,10)HOME,CLEAR
10    FORMAT('+',2A1)
      RETURN
      END
```

Reproduced from best available copy.

Page 1

```
      SUBROUTINE DFAULT
      REAL MWPO
      COMMON /INITAL/UA,TA,RH,CHAP,AL1,AL2,AL3,DPO,
    1 UO,DEGO,TPO,MWPO,CHPP,CWPP,RHO,ACO,RG1,S1,T1,
    2 TM1,RG2,S2,T2,TM2
      UA=5.0
      TA=20.0
      RH=50.0
      CHAP=00.0
      AL1=0.11
      AL2=0.6
      AL3=0.0
      DPO=4.0
      UO=20.0
      DEGO=90.0
      TPO=200.0
      MWPO=29.00
      CHPP=10.0
      CWPP=5.0
      RHO=1.0
      ACO=2.0
      RG1=0.07
      S1=1.3
      T1=1.0E6
      TM1=0.0
      RG2=1.0
      S2=1.8
      T2=3.0E3
      TM2=0.0
      RETURN
      END
      SUBROUTINE DISPLY
      RETURN
      END
      SUBROUTINE CLS
      LOGICAL A,B
      A=X'1C'
      B=X'1F'
      WRITE(3,10)A,B
 10   FORMAT('+',2A1)
      RETURN
      END
```

CHAPTER 6

IN-STACK OPACITY CALCULATOR

Program Author: Dr. P. A. Lawless

REVISION 1.2

ABSTRACT

This program calculates the in-stack plume opacity and mass extinction coefficient for a lognormal particle size distribution. Extinction efficiency factors can be computed using any one of several theoretical approximations at user-specified particle diameters. Default values have been supplied for all parameters and may be changed within reasonable bounds.

USER BENEFITS

- The in-stack plume opacity of emissions from pollution control equipment as measured by a transmissometer can be estimated with this program.

HARDWARE/SOFTWARE REQUIRED

This program runs on a TRS-80 Model I, III, or 4 microcomputer with two 5-inch floppy disks and 48K RAM memory. A printer is optional but recommended.

The programming code is Microsoft FORTRAN. A compiler and linking loader will be required if the source code is to be modified. The NEWDOS/80 Version 2 operating system has been used with the program.

METHOD

Plume opacity is often used as a criterion in regulating stationary air pollution sources. The program computes the opacity of the flue gas within the smoke stack. Under conditions where the particles in the flue are not modified at the point of emission, in-stack opacity is a good indicator of exterior opacity. Opacity is related to light transmittance by the following:

$$Op = 100(1-I/I_0) . \qquad (6-1)$$

Light transmittance is measured with in-stack transmissometers and can be computed with knowledge of the flue gas properties by the Bouger-Beer law:

$$I/I_o = \exp(-b_{ext}\ L) \qquad (6\text{-}2)$$

where

b_{ext} = extinction coefficient
L = pathlength (stack diameter) .

The extinction coefficient b_{ext} is a function of the particle concentration, size distribution, refractive index, and the wavelength of light. The extinction coefficient is given by

$$b_{ext} = Q(\pi\ D^2/4)N(D)\ dD \qquad (6\text{-}3)$$

where

Q = light extinction efficiency factor (sometimes called the cross section)
D = particle diameter
$N(D)$ = particle number distribution by diameter.

The parameter Q depends on the particle diameter, refractive index, and light wavelength. The refractive index, n, is a complex number with the real part, n_1, describing the velocity of light in the material and the imaginary part, n_2, describing the absorption and conversion of light to heat. Typical materials such as fly ash have a real refractive index of 1.5 and absorbing refractive index of 0.001. The wavelength of light is usually assigned to 0.55 μm to represent the maximum sensitivity of the human eye.

Equation (6-2) can be arranged to include mass concentration by

$$\begin{aligned} I/I_o &= \exp[-(b_{ext}/M)M\ L] \\ &= \exp(-b_m\ M\ L) \end{aligned} \qquad (6\text{-}4)$$

where

b_m = mass extinction coefficient
M = mass concentration.

The mass extinction coefficient is the ratio of the extinction coefficient to the mass concentration and is useful for predicting plume opacity. A reciprocal group was proposed by Ensor and Pilat (1) for correlating stack transmissometer data to particle mass data.

The particle mass concentration may also be computed from fundamental properties. The mass concentration is given by

$$M = (\pi/6)\ \ D^3\ N(D)\ dD\ . \qquad (6\text{-}5)$$

In the program, Equations (6-3) and (6-5) are numerically integrated over a particle size distribution to compute the mass extinction coefficient, which in turn is used to compute opacity with Equations (6-4) and (6-1). In addition, the total number is computed with

$$N_{TOT} = \ N(D)\ dD\ . \qquad (6\text{-}6)$$

The light extinction efficiency factor, Q, if computed exactly, requires a long and tedious series of calculations. The derivation of the interaction of electromagnetic waves with dielectric spheres was first reported by Mie (2). However, a number of approximations are also available that can reduce the number of calculations with some decrease in accuracy. Six options for calculating Q are available in this program. They are the full Mie Theory, the Deirmendjian Approximation, Anomalous Diffraction/Deirmendjian, Anomalous Diffraction Only, Rayleigh/Absorption Limit, and a constant value of 2.0. The Mie Theory and Deirmendjian Approximation are described in detail by Cowen, Ensor, and Sparks (3). The Anomalous Diffraction/Deirmendjian method is described by Kerker (4) and the Anomalous Diffraction Only by Van de Hulst (5). The Rayleigh/Absorption Limit method uses the Rayleigh equation to compute the extinction efficiency factors (Q) for small particle sizes while the value for large particles (>10 μm) is set to 2.0.

The lognormal particle size distribution is assumed in the calculation. The integration over the size distribution is performed with the Romberg integration method. The extreme range of particle radius from 0.01 to 100 μm requires that the method be applied in several panels. The derivation of the Romberg method is described by Ralston and Wilf (6).

The integral is calculated with a series of successive approximations. The integration tolerance is the agreement required between each approximation specified for an acceptable result. If the agreement is greater than the tolerance, the computation continues.

GENERAL RESTRICTIONS

Although the program will deliver results for a wide variety of inputs, negative values should not be used for any of the input quantities, and zero values should not be used for quantities representing physical dimensions.

The full Mie Theory or the Deirmendjian Approximation should be used for accurate work. For particles much smaller than the wavelength of light, the Rayleigh equations should give satisfactory results. The Anomalous Diffraction Theory assumes a particle refractive index close to 1, so caution should be exercised when using the methods that incorporate this approximation. The Constant Cross Section option should be used only for very coarse particles.

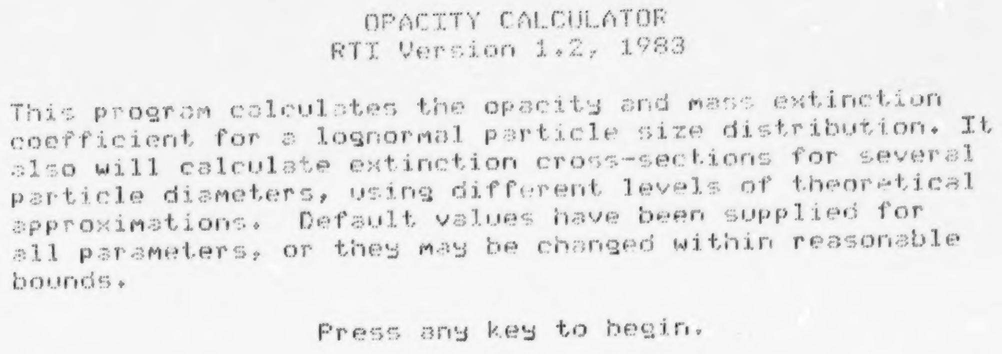
The Full Mie Theory calculations can be time-consuming; thus programs using this option may take a long time to run. A value of Q = 2.0 is used for all particles of diameter greater than 10 μm.

SAMPLE PROBLEM

Statement: Estimate the in-stack plume opacity from the emission properties of a sample gas given the geometric mean diameter, geometric standard deviation, particle density, particle concentration, refractive index, and stack diameter. In this case the default values are to be used.

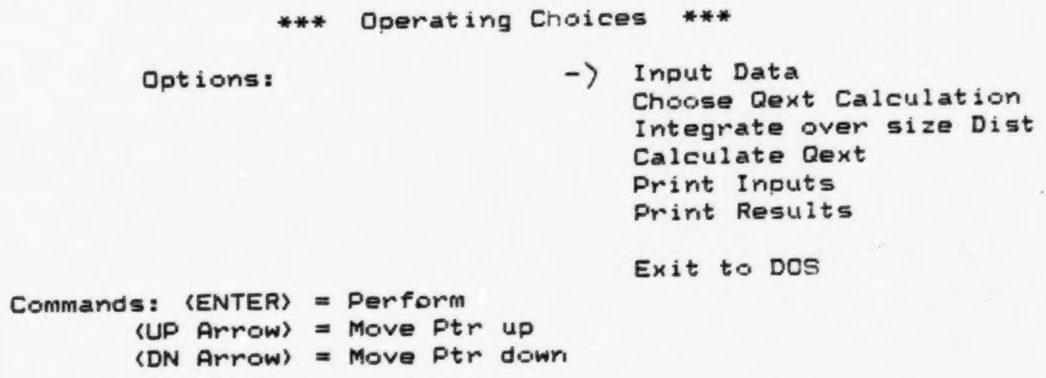
Results: This sample problem serves two purposes. First, the user can compare the results of the trial run with the numbers listed below to verify that the program is operating properly. Second, the step-by-step instructions will aid the operator in learning how to use the program.

When the program begins execution, the title of the program will appear along with a statement of what the program does (Figure 6-1). After the user pushes any key, the Main Menu, "Operating Choices," will appear as in Figure 6-2. The first two items of this menu allow data entry and choice of calculation method, the middle two items are used to initiate calculations, and the last two permit the user to view and print the data. To select an option, position the pointer in front of the choice and press <ENTER>. Since the first step in running a program is usually to input data, the user should choose the "Input Data" option causing the screen to appear as shown in Figure 6-3. Table 6-1 gives the meanings and units for the items in this menu.



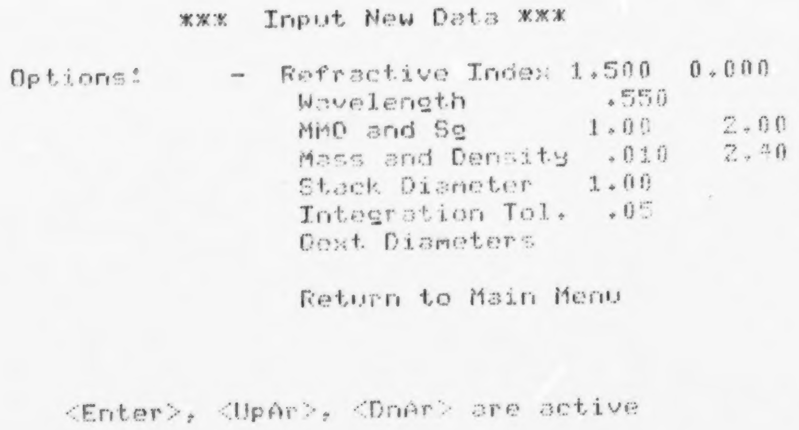
```
                    OPACITY CALCULATOR
                   RTI Version 1.2, 1983

This program calculates the opacity and mass extinction
coefficient for a lognormal particle size distribution. It
also will calculate extinction cross-sections for several
particle diameters, using different levels of theoretical
approximations.  Default values have been supplied for
all parameters, or they may be changed within reasonable
bounds.

                   Press any key to begin.
```

Figure 6-1. Title page.



```
              ***  Operating Choices  ***

      Options:                   -)  Input Data
                                     Choose Qext Calculation
                                     Integrate over size Dist
                                     Calculate Qext
                                     Print Inputs
                                     Print Results

                                     Exit to DOS
Commands: (ENTER) = Perform
       (UP Arrow) = Move Ptr up
       (DN Arrow) = Move Ptr down
```

Figure 6-2. Main menu for instack opacity program.



```
          ***  Input New Data ***

Options:     -  Refractive Index 1.500  0.000
                 Wavelength          .550
                 MMD and Sg        1.00     2.00
                 Mass and Density  .010     2.40
                 Stack Diameter   1.00
                 Integration Tol.  .05
                 Qext Diameters

                 Return to Main Menu


  <Enter>, <UpAr>, <DnAr> are active
```

Figure 6-3. Input data menu.

Reproduced from
best available copy.

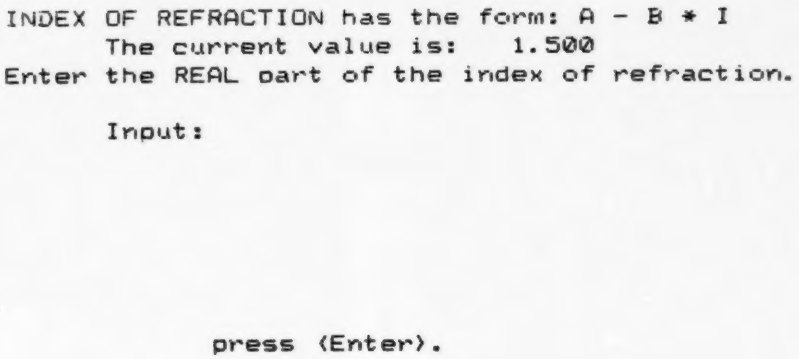
TABLE 6-1. UNITS FOR INPUT DATA

Menu entry	Definition	Units
Wavelength	Wavelength of light	μm
MMD	Mass median diameter	μm
Sg	Geometric standard deviation	dimensionless
Mass	Mass concentration	g/m^3
Density	Particle density	g/m^3
Stack diameter		meters
Integration tol.	Integration tolerance	dimensionless
Qext Diameters		μm

To change the refractive index from the observed default value, press <ENTER>. The resulting Figure, 6-4a, allows the user to enter a value for the real part. Note that pressing <D> will cause the program to use the current value listed on this page; therefore, press <D> for the sample problem. After <ENTER> or <D> is pressed, the user will be allowed to enter a value for the imaginary part on a screen appearing as Figure 6-4b. Press <D> again. The other data items listed on this page can be changed in a similar manner. Note that prolonged depression of the <D> key will result in the program skipping the second page of an option and returning to the "Input" menu. If this should occur merely push <ENTER> to return to the same option and depress the <D> key quickly to allow entry into the second page of data input.

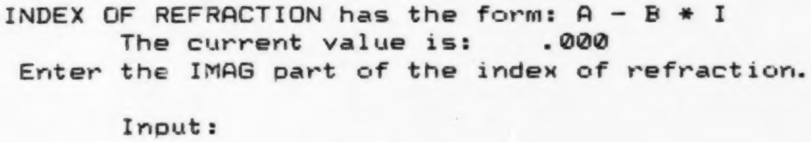
After the data are entered, or, as in this sample problem, examined, return to the Main Menu and designate the "Choose Qext Calculation" option. This will result in the screen appearing as shown in Figure 6-5. These are the six choices available for the calculation of Qext. They are listed in order of decreasing accuracy. At this point the user should choose the "Full Mie" option. Pushing <ENTER> after positioning the pointer will indicate the choice and return the "Operating Choices" menu to the screen. Next, the specific extinction coefficient and the percent opacity can be calculated by choosing the "Integrate over Size Dist" option. When the <ENTER> is pressed the screen will say "CALCULATING OPACITY," then intermediate results will be flashed periodically. When the calculations are complete (approximately 2.5 minutes) the screen will return to the "Operating Choices" menu. Input Data used by the program can be seen by selecting "Print Inputs." See Figure 6-6. To print this page, press <P>. To view or print the results, choose "Print Results." In response, Figure 6-7 will appear. On this page "Scattering Approximation" refers to the choice of calculation method for Qext. The number listed next to it corresponds to the number in the choice list. In this case the 1 indicates that the "Full Mie" option was selected.

The last option of this program is "Calculate Qext." When the user has input all pertinent data and has chosen a Qext calculation method, this option should be employed. The screen will say "Calculating Qext." When the calculations are complete the Main Menu will return. At this point



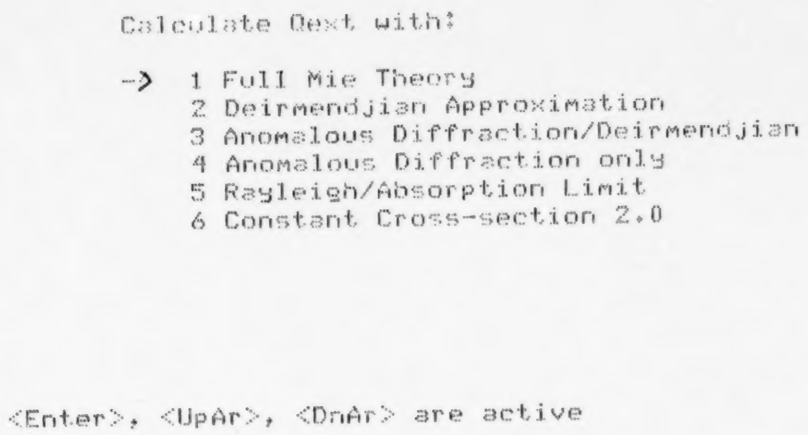
```
INDEX OF REFRACTION has the form: A - B * I
      The current value is:   1.500
Enter the REAL part of the index of refraction.

      Input:




            press <Enter>.
```

Figure 6-4a. Entry page for the real part of the refractive index.



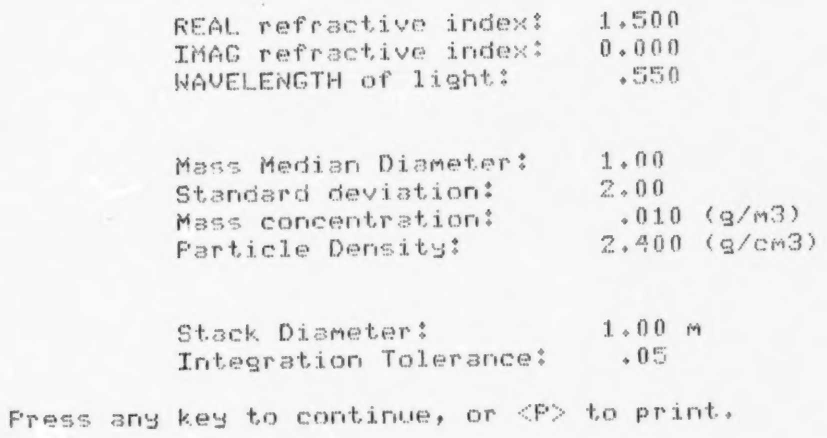
```
INDEX OF REFRACTION has the form: A - B * I
      The current value is:    .000
 Enter the IMAG part of the index of refraction.

      Input:
```

Figure 6-4b. Entry page for the imaginary part of the refractive index.



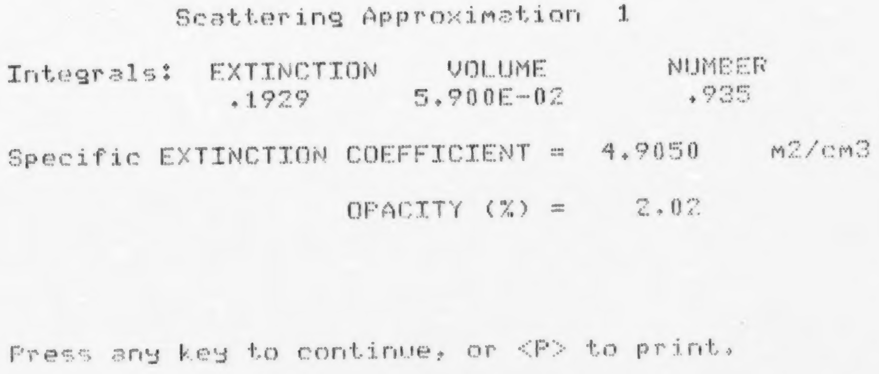
```
        Calculate Qext with:

        ->  1 Full Mie Theory
            2 Deirmendjian Approximation
            3 Anomalous Diffraction/Deirmendjian
            4 Anomalous Diffraction only
            5 Rayleigh/Absorption Limit
            6 Constant Cross-section 2.0




  <Enter>, <UpAr>, <DnAr> are active
```

Figure 6-5. "Choose Qext Calculation" menu.



```
       REAL refractive index:    1.500
       IMAG refractive index:    0.000
       WAVELENGTH of light:       .550


       Mass Median Diameter:     1.00
       Standard deviation:       2.00
       Mass concentration:        .010 (g/m3)
       Particle Density:         2.400 (g/cm3)


       Stack Diameter:           1.00 m
       Integration Tolerance:     .05

  Press any key to continue, or <P> to print.
```

Figure 6-6. Input data output for "Integrate . . ." option.



```
          Scattering Approximation  1

Integrals:  EXTINCTION     VOLUME        NUMBER
              .1929      5.900E-02        .935

Specific EXTINCTION COEFFICIENT =  4.9050     m2/cm3

                    OPACITY (%) =     2.02



Press any key to continue, or <P> to print.
```

Figure 6-7. Results output for "Integrate . . ." option.

choosing "Print Inputs" will result in the screen appearing as Figure 6-8 (pressing <P> will yield an identical printout); "Print Results" will yield Figure 6-9.

This ends the sample problem. At this point the user should return to one of the first two options "Input Data" or "Choose Qext Calculation" and change values in order to examine the response of the program.

REFERENCES

1. Ensor, D. S., and M. J. Pilat. Calculation of Smoke Plume Opacity from Particulate Air Pollutant Properties. J. Air Poll. Contr. Assn., 21:496, 1971.

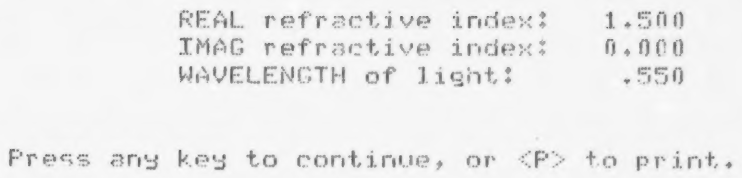
2. Mie, G. Ann. Physik 25:377. 1908

3. Cowen, S. J., D.S. Ensor, and L. E. Sparks. TRS-80 In-Stack Opacity Computer Programs: User and Programmer Manual. Report prepared under Cooperative Agreement No. R-806718010 for U.S. Environmental Protection Agency, Research Triangle Park, North Carolina. 1984.

4. Kerker, M. The Scattering of Light. Academic Press, New York, 1969.

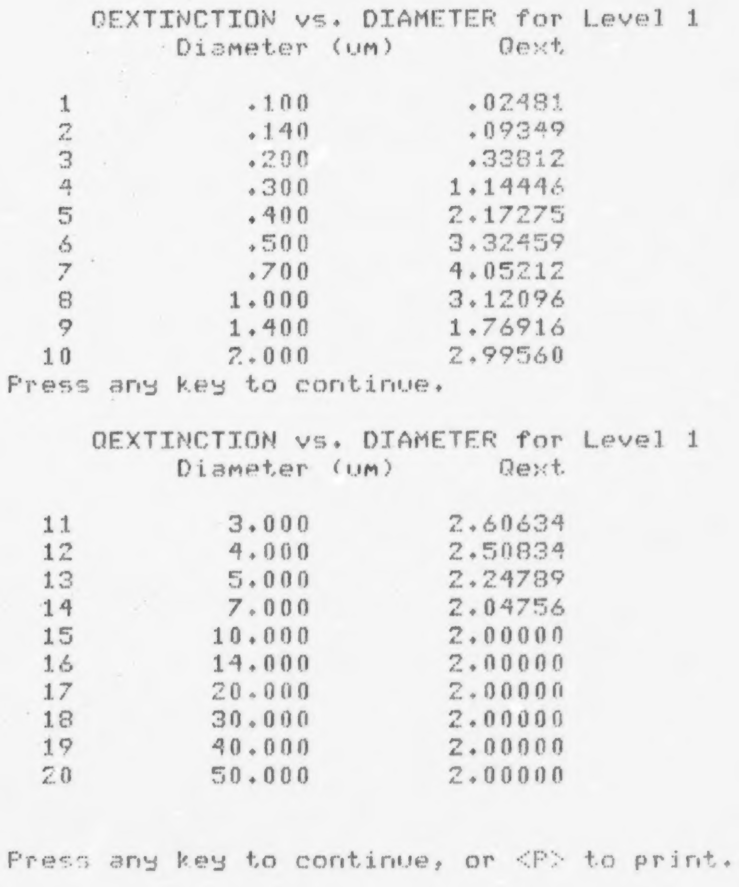
5. Van de Hulst, H. C. Light Scattering by Small Particles. Dover, New York, reprint of Wiley and Sons 1957 publication. 1981.

6. Ralston, A., and H. S. Wilf (eds.). Mathematical Methods for Digital Computers. Wiley and Sons, New York, 1967.



```
              REAL refractive index:    1.500
              IMAG refractive index:    0.000
              WAVELENGTH of light:       .550


      Press any key to continue, or <P> to print.
```

Figure 6-8. Input data output for "Calculate Qext" option.



```
     QEXTINCTION vs. DIAMETER for Level 1
           Diameter (um)        Qext

   1           .100          .02481
   2           .140          .09349
   3           .200          .33812
   4           .300         1.14446
   5           .400         2.17275
   6           .500         3.32459
   7           .700         4.05212
   8          1.000         3.12096
   9          1.400         1.76916
  10          2.000         2.99560
Press any key to continue.

     QEXTINCTION vs. DIAMETER for Level 1
           Diameter (um)        Qext

  11          3.000         2.60634
  12          4.000         2.50834
  13          5.000         2.24789
  14          7.000         2.04756
  15         10.000         2.00000
  16         14.000         2.00000
  17         20.000         2.00000
  18         30.000         2.00000
  19         40.000         2.00000
  20         50.000         2.00000


Press any key to continue, or <P> to print.
```

Figure 6-9. Results output for "Calculate Qext" option.

```
      PROGRAM OPRUN

C THIS PROGRAM IS THE DRIVER PROGRAM FOR OBTAINING
C AND COMPUTING OPACITY DATA

      INTEGER POINTR,COLUMN,ELMNTS
      REAL MR,MI,M
      COMMON /APP/IAPP
      COMMON /ROM/MR,MI,W,SG,DG,M,D,S,ER,DP,D1,D2,
     * SP,T1,T2,T3,ICT,V1,V2,V3,OP
      COMMON /DIAQ/DQ(20),QX(20)
      COMMON /DIA/DI(10)

C PUT UP THE SCREEN MESSAGE

      CALL SCRNMS

      IPRT = 1
      POINTR = 1

1100  CONTINUE

      CALL CLS

      ELMNTS = 8
      COLUMN = 35

      WRITE(1,102)
 102  FORMAT('+',17X,'***  Operating Choices  ***'/
     * /'+',9X,'Options:',21X,'Input Data'
     * /'+',38X,'Choose Qext Calculation'
     * /'+',38X,'Integrate over size Dist'
     * /'+',38X,'Calculate Qext'
     * /'+',38X,'Print Inputs'
     * /'+',38X,'Print Results'
     * /'+'
     * /'+',38X,'Exit to DOS'
     * /'+',1X,'Commands: <ENTER> = Perform'
     * /'+',8X,'<UP Arrow> = Move Ptr up'
     * /'+',8X,'<DN Arrow> = Move Ptr down')

      CALL MENU(ELMNTS,COLUMN,POINTR)

      GO TO(1300,1500,1700,1900,2100,2300,
     * 1100,2500),POINTR

1300  CALL DTI
      GO TO 1100

1500  CALL CHOOSE
      GO TO 1100
```

```
1700    CALL OPCITY
        IPRT = 1
        GO TO 1100

1900    CALL QXCALC
        IPRT = 2
        GO TO 1100

2100    IF(IPRT.EQ.1) CALL PRTIN1
        IF(IPRT.EQ.2) CALL PRTIN2
        GO TO 1100

2300    IF(IPRT.EQ.1) CALL PRTRS1
        IF(IPRT.EQ.2) CALL PRTRS2
        GO TO 1100

2500    CALL CLS
        STOP

        END
```

```
          SUBROUTINE DTI

C THIS PROGRAM IS A SECONDARY MENU PROGRAM FOR INPUTTING
C DATA

         INTEGER POINTR,COLUMN,ELMNTS
         REAL MR,MI,M
         COMMON /ROM/ MR,MI,W,SG,DG,M,D,S,ER,DP,D1,D2,
     *   SP,T1,T2,T3,ICT,V1,V2,V3,OP

         POINTR = 1

1100     CONTINUE

C CLEAR THE SCREEN

         CALL CLS

         ELMNTS = 9
         COLUMN = 23

         WRITE(1,1) MR,MI,W,DG,SG,M,D,S,ER
    1    FORMAT('+',19X,'***  Input New Data ***'/
     *   /'+',9X,'Options:',9X,'Refractive Index ', F5.3, 2X,F5.3
     *   /'+',26X,'Wavelength',7X,F5.3
     *   /'+',26X,'MMD and Sg',6X,F5.2,3X,F5.2
     *   /'+',26X,'Mass and Density',1X,F5.3,2X,F5.2
     *   /'+',26X,'Stack Diameter',2X,F5.2
     *   /'+',26X,'Integration Tol.',F5.2
     *   /'+',26X,'Qext Diameters'
     *   /'+'
     *   /'+',26X,'Return to Main Menu')

         CALL CBL

         WRITE(1,2)
    2    FORMAT('+',12X,'<Enter>, <UpAr>, <DnAr> are active')

         CALL MENU(ELMNTS,COLUMN,POINTR)

         GO TO(1300,1500,1700,1900,2100,2300,2500,
     *   1100,2700),POINTR

1300     CALL DATIN(1)
         GO TO 1100

1500     CALL DATIN(2)
         GO TO 1100

1700     CALL DATIN(3)
         GO TO 1100

1900     CALL DATIN(4)
         GO TO 1100
```

Reproduced from best available copy.

```
2100    CALL DATIN(5)
        GO TO 1100

2300    CALL DATIN(6)
        GO TO 1100

2500    CALL DIAIN
        GO TO 1100

2700    RETURN

        END


        SUBROUTINE CHOOSE

C THIS PROGRAM IS A SECONDARY MENU PROGRAM FOR CHOOSING
C THE LEVEL OF APPROXIMATION USED

        INTEGER POINTR,COLUMN,ELMNTS
        COMMON /APP/IAPP
        DATA POINTR/1/

C CLEAR THE SCREEN

        CALL CLS

        ELMNTS = 6
        COLUMN = 20

        WRITE(1,1)
   1    FORMAT('+',19X.'Calculate Qext with:'/
     *  /'+',23X.'1 Full Mie Theory'
     *  /'+',23X,'2 Deirmendjian Approximation'
     *  /'+',23X,'3 Anomalous Diffraction/Deirmendjian'
     *  /'+',23X,'4 Anomalous Diffraction only'
     *  /'+'.23X,'5 Rayleigh/Absorption Limit'
     *  /'+',23X,'6 Constant Cross-section 2.0')

        CALL CBL

        WRITE(1,2)
   2    FORMAT('+',12X,'<Enter>, <UpAr>, <DnAr> are active')

        CALL MENU(ELMNTS,COLUMN,POINTR)

        IAPP = POINTR

        RETURN

        END
```

Reproduced from best available copy.

```
      SUBROUTINE DATIN(IN)

C DATA ENTRY ROUTINE FOR OPACITY PROGRAM

      LOGICAL TOF
      REAL MR,MI,M
      COMMON /ROM/ MR,MI,W,SG,DG,M,D,S,ER,DP,D1,D2,
     * SP,T1,T2,T3,ICT,V1,V2,V3,OP
      DATA MR/1.5/MI/0.0/W/0.55/SG/2.0/DG/1.0/M/0.01/
      DATA D/2.4/S/1.0/ER/0.05/
      DATA TOF/X'1C'/

      CALL CLS

      GO TO (100,300,500,700,900,1100),IN

100   CALL CBL
      WRITE(1,5) TOF,MR
5     FORMAT('+',9X,'To use the default value, press <D>.',A1,
     * 9X,'INDEX OF REFRACTION has the form: A - B * I'/
     * '+',15X,'The current value is:',F8.3/
     * '+'9X,'Enter the REAL part of the index of refraction.'/)

      CALL DATGET(MR,3.0,1.0)

      WRITE(1,6) TOF,MI
6     FORMAT('+',A1,9X,
     * 'INDEX OF REFRACTION has the form: A - B * I'/
     * '+',15X,'The current value is:',F8.3/
     * '+',9X,'Enter the IMAG part of the index of refraction.'/)

      CALL DATGET(MI,10.,-1.0)

      RETURN

300   CALL CBL
      WRITE(1,7) TOF,W
7     FORMAT('+',9X,'To use the default value, press <D>.',A1,
     * 9X,'WAVE LENGTH OF LIGHT has the units of microns.'/
     * '+',15X,'The current value is:',F8.3/
     * '+',9X,'Enter the WAVELENGTH.'/)

      CALL DATGET(W,1.0,0.1)

      RETURN
```

```
500     CALL CBL
        WRITE(1,8) TOF,DG
  8     FORMAT('+',9X,'To use the default value, press <D>.',A1,
     *  9X,'MASS MEDIAN DIAMETER has the units microns.'/
     *  '+',15X,'The current value is:',F8.3/
     *  '+',9X,'Enter the MASS MEDIAN DIAMETER.'/)

        CALL DATGET(DG,60.,0.01)

        WRITE(1,9) TOF,SG
  9     FORMAT('+',A1,9X,
     *  'GEOMETRIC STANDARD DEVIATION is dimensionless.'/
     *  '+',15X,'The current value is:',F8.3/
     *  '+',9X,'Enter the GEOMETRIC STANDARD DEVIATION.'/)

        CALL DATGET(SG,20.,1.01)

        RETURN

700     CALL CBL
        WRITE(1,10) TOF,M
 10     FORMAT('+',9X,'To use the default value, press <D>.',A1,
     *  9X,'MASS CONCENTRATION has the units g/m3.'/
     *  '+',15X,'The current value is:',F8.3/
     *  '+',9X,'Enter the MASS CONCENTRATION.'/)

        CALL DATGET(M,20.,0.001)

        WRITE(1,11) TOF,D
 11     FORMAT('+',A1,9X,
     *  'PARTICLE DENSITY has the units g/cm3. '/
     *  '+',15X,'The current value is:',F8.3/
     *  '+',9X,'Enter the PARTICLE DENSITY.'/)

        CALL DATGET(D,5.0,0.5)

        RETURN

900     CALL CBL
        WRITE(1,12) TOF,S
 12     FORMAT('+',9X,'To use the default value, press <D>.',A1,
     *  9X,'STACK DIAMETER has the units of meters.'/
     *  '+',15X,'The current value is:',F8.3/
     *  '+',9X,'Enter the STACK DIAMETER.'/)

        CALL DATGET(S,20.,0.1)

        RETURN
```

```
1100    CALL CBL
        WRITE(1,13) TOF,ER
  13    FORMAT('+',9X,'To use the default value, press <D>.',A1,
     *  9X,'INTEGRATION TOLERANCE is dimensionless.'/
     *  '+',15X,'The current value is:',F8.3//
     *  '+A value smaller than 0.05 will require MUCH time!'//
     *  '+',9X,'Enter the INTEGRATION TOLERANCE.'/)

        CALL DATGET(ER,1.0,0.001)

        RETURN

        END



        SUBROUTINE DIAIN

        COMMON /DIAQ/ DQ(20),QX(20)
        DATA DQ/0.1,0.14,0.2,0.3,0.4,0.5,0.7,1.0,1.4,
     *  2.,3.,4.,5.,7.,10.,14.,20.,30.,40.,50./
        DATA QX/20*2.0/

        DO 200 I = 1,20

        CALL CLS

        WRITE(1,1) I,DQ(I)
   1    FORMAT (///'+',10X,'Qext Diameter has the units microns.'/
     *  '+',10X,'The current value for #',I2,' is:',F8.3/
     *  '+',10X,'Enter the new  Qext Diameter, or <D>'/)

        CALL DATGET(DQ(I),60.,0.05)

 200    CONTINUE

        RETURN

        END
```

```
      SUBROUTINE OPCITY

C PROGRAM TO integrate plume OPACITY USING EXACT MIE LIGHT SCATTERING
C EQUATIONS. A LOG-NORMAL PARTICLE SIZE DISTRIBUTION IS USED.
C VERSION 1.1 WRITTEN BY S. J. COWEN OF METEOROLOGY RESEARCH INC.
C ALTADENA, CALIF. AND D. S. ENSOR OF RESEARCH TRIANGLE
C INSTITUTE, RTP, NC.
C ADAPTED BY P. A. LAWLESS, RTI 01/25/83

      REAL MR,MI,M
      COMMON /DIA/DI(10)
      COMMON /ROM/MR,MI,W,SG,DG,M,D,S,ER,DP,D1,D2,
    * SP,T1,T2,T3,ICT,V1,V2,V3,OP
      DATA DI/0.01,0.1,0.3,0.5,1.,3.,5.,10.,30.,100./
      DATA SP,OP/0.0,0.0/

C          DEFINITIONS

C    MR    REAL PART OF INDEX OF REFRACTION
C    MI    IMAG PART OF INDEX OF REFRACTION
C    W     WAVELENGTH OF LIGHT
C    SG    GEOMETRIC STANDARD DEVIATION
C    DG    GEOMETRIC MASS MEAN DIAMETER
C    DP    PARTICLE DIAMTER (VAR. OF INTEG.)

C  COMPUTATIONS

      CALL CLS

      WRITE(1,2)
  2   FORMAT (///////'+',11X,'C A L C U L A T I N G   O P A C I T Y')

      V1 = 0.0
      V2 = 0.0
      V3 = 0.0

      DO 200 NN = 1,9
      N1=NN+1
      D1=DI(NN)
      D2=DI(N1)

C CALCULATE THE INTEGRANDS

      CALL ROMBRG

      V1=V1+T1
      V2=V2+T2
      V3=V3+T3
```

```
        CALL CLS

        WRITE(1,7) D1,D2
   7    FORMAT('+',13X,'Lower diameter - Upper diameter'/
     *  '+',14X,F8.2,10X,F8.2/)
        WRITE(1,3) T1,T2,T3,ICT
   3    FORMAT ('+            Extinction   Volume   Number   # Terms'/
     *  '+',12X,3(1PG10.2),0PI6/)
        WRITE(1,4) V1,V2,V3
   4    FORMAT ('+Cumulative  Extinction   Volume   Number '/
     *  '+',12X,3(1PG10.2)/)

        CALL CBL

        WRITE(1,5)
   5    FORMAT('+',10X,'Hold <Break> to abort calculation.')

        IF(INKEY(0).EQ.1) RETURN

 200    CONTINUE

C    CALCULATE SP AND OP

        SP = 1.5*V1/V2
        OP = 100.0*(1.0 - EXP(-SP*M*S/D))

        RETURN

        END
```

```
      SUBROUTINE QXCALC

      REAL MR,MI,M
      COMMON /DIAQ/DQ(20),QX(20)
      COMMON /ROM/MR,MI,W,SG,DG,M,D,S,ER,DP,D1,D2,
     * SP,T1,T2,T3,ICT,V1,V2,V3,OP
      DATA PI/3.14159265/

C     MR      REAL PART OF INDEX OF REFRACTION
C     MI      IMAG PART OF INDEX OF REFRACTION
C     W       WAVELENGTH OF LIGHT
C     DQ      PARTICLE DIAMTER (VAR. OF INTEG.)

C   COMPUTATIONS

      CALL CLS

      WRITE(1,2)
   2  FORMAT (////////'+',13X,'C A L C U L A T I N G   Q E X T ')
      CALL.QCAL(20)

      DO 200 I = 1,20

      CALL PLOT(I)

      XI = PI*DQ(I)/W
      QX(I) = QCALC(XI,MR,MI)

 200  CONTINUE

      RETURN

      END
```

239

```
      SUBROUTINE ROMBRG

C   ROMBERG INTEGRATION SUBPROGRAM

      REAL MR,MI,M,I1,I2
      DIMENSION W1(10,10),W2(10,10),W3(10,10)
      COMMON /ROM/MR,MI,W,SG,DG,M,D,S,ER,DP,D1,D2,
   *  SP,T1,T2,T3,ICT,V1,V2,V3,OP
      DATA PI/3.14159265/

      H=D2-D1
      DP=D1
      XI = PI*DP/W

      Q = QCALC(XI,MR,MI)

      F = FRACT(DG,SG,DP)

      T1=DP*DP*F*.5
      I1=T1*Q
      I2=T1*DP
      I3=F

      DP=D2
      XI = PI*DP/W

      Q = QCALC(XI,MR,MI)

      F = FRACT(DG,SG,DP)

      T1=DP*DP*F*.5
      F1=T1*Q
      F2=T1*DP

      W1(1,1)=((I1+F1)*H)*.5
      W2(1,1)=((I2+F2)*H)*.5
      W3(1,1)=((I3+F)*H)*.5
      ICT = 2

C K CAN ONLY GO UP TO 9 TO AVOID EXCEEDING ARRAY LIMITS

      DO 1000 K = 1,9
      H=H*.5
      S1=0.0
      S2=0.0
      S3=0.0
      IM=2**(K-1)

      CALL QCAL(IM)

      DO 200 J = 1,IM
      ICT=ICT+1

      CALL PLOT(J)
```

```
C CHECK FOR ABORT REQUEST.

        IF(INKEY(0).EQ.1) RETURN

        DP = D1 + H*(2*J - 1)
        XI = PI*DP/W

        Q = QCALC(XI,MR,MI)

        F = FRACT(DG,SG,DP)

        T1=DP*DP*F*.5
        S1=S1+T1*Q
        S2=S2+T1*DP
        S3=S3+F
 200    CONTINUE

        K1=K+1
        W1(K1,1)=W1(K,1)*.5+H*S1
        W2(K1,1)=W2(K,1)*.5+H*S2
        W3(K1,1)=W3(K,1)*.5+H*S3

        DO 400 L=1,K
        RL = 4**L
        LU=K+1-L
        L1=LU+1
        LV=L+1
        W1(LU,LV)=(RL+W1(L1,L)-W1(LU,L))/(RL-1)
        W2(LU,LV)=(RL*W2(L1,L)-W2(LU,L))/(RL-1)
        W3(LU,LV)=(RL*W3(L1,L)-W3(LU,L))/(RL-1)
 400    CONTINUE

        A1=ABS((W1(1,LV)-W1(1,K))/W1(1,LV))

        T1=W1(1,LV)
        T2=W2(1,LV)
        T3=W3(1,LV)
        IF(A1.LE.ER) GO TO 300
1000    CONTINUE

        WRITE(3,10)
  10    FORMAT('+','R O M B E R G   P R O B L E M')

        CALL DELAY

 300    RETURN

        END
```

```
      SUBROUTINE PRTIN1

      REAL MR,MI,M
      COMMON /DIAQ/DQ(20),QX(20)
      COMMON /ROM/ MR,MI,W,SG,DG,M,D,S,ER,DP,D1,D2,
     * SP,T1,T2,T3,ICT,V1,V2,V3,OP

      CALL CLS

      LUN = 1

100   WRITE(LUN,1) MR,MI,W
 1    FORMAT('+',10X,'REAL refractive index:',F8.3/
     * '+',10X,'IMAG refractive index:',F8.3/
     * '+',10X,'WAVELENGTH of light:',2X,F8.3//)

      WRITE(LUN,2) DG,SG,M,D
 2    FORMAT('+',10X,'Mass Median Diameter:',F8.2/
     * '+',10X,'Standard deviation:',2X,F8.2/
     * '+',10X,'Mass concentration:',5X,F6.3,' (g/m3)'/
     * '+',10X,'Particle Density:',5X,F8.3,' (g/cm3)'//)

      WRITE(LUN,3) S,ER
 3    FORMAT('+',10X,'Stack Diameter:',6X,F8.2,' m'/
     * '+',10X,'Integration Tolerance:',F7.2/)

      IF(LUN.EQ.2) RETURN
      IF(LUN.EQ.1) WRITE(LUN,4)
 4    FORMAT('+Press any key to continue, or <P> to print.')

      IASC = INKEY(1).AND.95

      IF(IASC.NE.80.) RETURN

      CALL BUSY

      LUN = 2
      GO TO 100

      END
      SUBROUTINE PRTIN2

      REAL MR,MI,M
      COMMON /DIAQ/DQ(20),QX(20)
      COMMON /ROM/ MR,MI,W,SG,DG,M,D,S,ER,DP,D1,D2,
     *  SP,T1,T2,T3,ICT,V1,V2,V3,OP

      CALL CLS

      LUN = 1

100   WRITE(LUN,1) MR,MI,W
  1   FORMAT('+',10X,'REAL refractive index:',F8.3/
     *  '+',10X,'IMAG refractive index:',F8.3/
     *  '+',10X,'WAVELENGTH of light:',2X,F8.3//)

      IF(LUN.EQ.2) RETURN
      IF(LUN.EQ.1) WRITE(LUN,4)
  4   FORMAT('+Press any key to continue, or <P> to print.')

      IASC = INKEY(1).AND.95

      IF(IASC.NE.80.) RETURN

      CALL BUSY

      LUN = 2
      GO TO 100

      END
      SUBROUTINE PRTRS1

      REAL MR,MI,M
      COMMON /APP/IAPP
      COMMON /DIAG/DQ(20),QX(20)
      COMMON /ROM/MR,MI,W,SG,DG,M,D,S,ER,DP,D1,D2,
     * SP,T1,T2,T3,ICT,V1,V2,V3,OP

      CALL CLS

      LUN = 1

100   WRITE(LUN,5) IAPP
  5   FORMAT('+',10X,'Scattering Approximation',I3/)

      WRITE(LUN,1) V1,V2,V3
  1   FORMAT('+Integrals:  EXTINCTION',4X,'VOLUME',6X,' NUMBER'/
     * '+',11X,1PG11.4,G11.3,2X,0PF9.3/)

      WRITE(LUN,2) SP
  2   FORMAT('+Specific EXTINCTION COEFFICIENT = ', 1PG11.5,
     * 'm2/cm3'/)

      WRITE(LUN,3) OP
  3   FORMAT('+',20X,'OPACITY (%) = ',F7.2//)

      IF(LUN.EQ.2) RETURN
      IF(LUN.EQ.1) WRITE(LUN,4)
  4   FORMAT(//'+Press any key to continue, or <P> to print.')

      IASC = INKEY(1).AND.95

      IF(IASC.NE.80.) RETURN

      CALL BUSY

      LUN = 2
      GO TO 100

      END
      SUBROUTINE PRTRS2

      REAL MR,MI,M
      COMMON /APP/IAPP
      COMMON /DIAQ/DQ(20),QX(20)
      COMMON /ROM/ MR,MI,W,SG,DG,M,D,S,ER,DP,D1,D2,
     * SP,T1,T2,T3,ICT,V1,V2,V3,OP

      LUN = 1

100   CONTINUE

      DO 200 I = 1,2

      CALL CLS

      WRITE(LUN,5) IAPP
  5   FORMAT('+',5X,'QEXTINCTION vs. DIAMETER for Level',I2 /
     * '+',10X,'Diameter (um)       Qext'/)

      DO 220 J = 1,10
      K = 10*(I-1) + J
220   WRITE(LUN,6) K,DQ(K),QX(K)
  6   FORMAT('+',2X,I2,F14.3,4X,F11.5)

      IF(LUN.EQ.2) WRITE(LUN,7)
  7   FORMAT(//)

      IF(LUN.EQ.2.OR.I.EQ.2) GO TO 200

      WRITE(LUN,8)
  8   FORMAT('+Press any key to continue.')

      IASC = INKEY(1)

200   CONTINUE

      IF(LUN.EQ.2) RETURN
      IF(LUN.EQ.1) WRITE(LUN,4)
  4   FORMAT(//'+Press any key to continue, or <P> to print.')

      IASC = INKEY(1).AND.95

      IF(IASC.NE.80.) RETURN

      CALL BUSY

      LUN = 2
      GO TO 100

      END
      SUBROUTINE DATGET(X,VU,VW)

C THIS SUBROUTINE PERFORMS A REPETITIVE DATA ROUTINE AND
C RETURNS WITH A VALUE FOR X.  ON ENTERING THIS ROUTINE, X
C CONTAINS THE PRESENT (DEFAULT) VALUE.

      LOGICAL A
      INTEGER DADR,DIGITS
      DATA DADR,DIGITS /X'3801',X'3830'/

      WRITE(1,3)
    3 FORMAT('+',15X,'Input: ')

  100 A = PEEK(DADR)

      IF(A.EQ.16) GO TO 300

      A = PEEK(DIGITS)

      IF(A.EQ.0) GO TO 100

      READ(1,1) VL
    1 FORMAT(F10.0)

      CALL TEST(VL,VU,VW)

      X = VL

  300 RETURN

      END

      SUBROUTINE SCRNMS

C CLEAR SCREEN AND DISPLAY MESSAGE

      CALL CLS

      WRITE(1,1)
    1 FORMAT(/'+',21X,'OPACITY CALCULATOR'/'+',19X,
     * 'RTI Version 1.2, 1983'//
     * '+This program calculates the opacity and mass extinction'/
     * '+coefficient for a lognormal particle size distribution. It'/
     * '+also will calculate extinction cross-sections for several'/
     * '+particle diameters, using different levels of theoretical'/
     * '+approximations.  Default values have been supplied for'/
     * '+all parameters, or they may be changed within reasonable'/
     * '+bounds.'//
     * 19X,'Press any key to begin.')

      IASC = INKEY(1)

      RETURN
      END
      FUNCTION QCALC(XI,MR,MI)

C    SUBROUTINE TO CALCULATE Q USING THE MIE EQUATIONS, THE
C    DEIRMENDJIAN APPORXIMATIONS, OR ANOMALOUS DIFFRACTION

      REAL MR,MI
      COMMON /APP/IAPP

      RHO = XI*2.0*(MR - 1.0)

 100  GO TO (300,500,700,900,1100),IAPP
      QCALC = 2.0
      RETURN

 300  QCALC = QMIE(XI,MR,MI)
      RETURN

 500  TG = MI/(MR - 1.0)
      G = ATAN(TG)
      Q = QDEI(RHO,G,TG)
      F = DFACT(RHO,MR,TG)
      QCALC = Q*F
      RETURN

 700  Q = QAD(RHO)
      F = DFACT(RHO,MR,0.0)
      QCALC = Q*F
      RETURN

 900  QCALC = QAD(RHO)
      RETURN

1100  QCALC = QRA(XI)
      RETURN

      END
      FUNCTION QMIE(XI,MR,MI)

C   SUBROUTINE TO CALCULATE Q USING THE MIE EQUATIONS
C   THE ARRAYS ARE ADJUSTED TO THE MAXIMUM NUMBER OF TERMS

      REAL MR,MI
      DIMENSION WR(82),WI(82),AR(82),AI(82),ER(82),
     * FR(82)
      DATA WR,WI,AR,AI,ER,FR/492*0.0/

C   USE A VALUE OF 2.0 FOR Q THE LIMIT FOR LARGE XI

      QMIE = 2.0
      IF(XI.GT.40.0) RETURN

      N = 1.0*XI
      N = MIN0(N,80)
      N = MAX0(N,3)
      IN = N + 2

C  PREPARE 0 AND -1 SUBSCRIPTS FOR THE BESSEL FUNCTIONS

      WR(1) = COS(XI)
      WI(1) = -SIN(XI)
      WR(2) = -WI(1)
      WI(2) = WR(1)

C CALCULATE THE BESSEL FUNCTIONS BY RECURSION

      DO 200 I = 3,IN
      I1 = I-1
      I2 = I-2
      PR = (2*I2 - 1)/XI
      WR(I) = PR*WR(I1) - WR(I2)
 200  WI(I) = PR*WI(I1) - WI(I2)

C  CALCULATE THE An(0) TERMS

      PP = MR*XI
      QQ = MI*XI
      IF(QQ.GT.85.) GO TO 100

C  SINH(QQ) AND COSH(QQ)

      TT = EXP(QQ)
      SH = (TT-1.0/TT)*.5
      CH = (TT+1.0/TT)*.5
      T1 = SIN(PP)
      T2 = T1*T1 + SH*SH
      AR(2) = T1*COS(PP)/T2
      AI(2) = SH*CH/T2
      GO TO 300
```

```
C TAKEN FROM DAVID'S PAPER

 100      AR(2) = 0.0
          AI(2) = 1.0

C CALCULATE THE COMPLEX INDEX SUARE MAGNITUDE.

 300      TT = MR*MR + MI*MI

C CALCULATE THE An TERMS

          DO 400 I = 3,IN
          I1 = I-1
          I2 = I-2
          PR = I2/XI
          QR = (PR*MR)/TT
          QI = (PR*MI)/TT
          PR = QR - AR(I1)
          PI = QI - AI(I1)
          TX = PR*PR + PI*PI
          AR(I) = PR/TX - QR
          AI(I) = -PI/TX - QI
 400      CONTINUE

C CALCULATE THE an TERMS

          DO 600 I = 3,IN
          I1 = I - 1
          I2 = I - 2
          TR = (AR(I)*MR - AI(I)*MI)/TT + I2/XI
          TI = (AI(I)*MR + AR(I)*MI)/TT
          PR = TR*WR(I) - WR(I1)
          PI = TI*WR(I)
          QR = TR*WR(I) - TI*WI(I) - WR(I1)
          QI = TR*WI(I) + TI*WR(I) - WI(I1)
          TX = QR*QR + QI*QI
          ER(I) = (PR*QR + PI*QI)/TX
 600      CONTINUE

C CALCULATE THE bn TERMS

          DO 800 I = 3,IN
          I1 = I - 1
          I2 = I - 2
          TR = MR*AR(I) + MI*AI(I) + I2/XI
          TI = MR*AI(I) - MI*AR(I)
          UR = TR*WR(I) - WR(I1)
          UI = TI*WR(I)
          QR = TR*WR(I) - TI*WI(I) - WR(I1)
          PI = TR*WI(I) + TI*WR(I) - WI(I1)
          TX = QR*QR + PI*PI
          FR(I) = (UR*QR + UI*PI)/TX
 800      CONTINUE
```

```
C SUM THE POSITIVE AND NEGATIVE PARTS OF THE ER's and FR's,
C SEPARATELY

          SPE = 0.0
          SPF = 0.0
          SME = 0.0
          SMF = 0.0

          DO 1000 IR = 3,IN
          I = IN + 3 - IR
          I2 = 2*(I - 2) + 1

          IF(ER(I).LT.0.0) GO TO 1100
          SPE = SPE + I2*ER(I)
          GO TO 1300

1100      SME = SME + I2*ER(I)

1300      IF(FR(I).LT.0.0) GO TO 1500
          SPF = SPF + I2*FR(I)
          GO TO 1000

1500      SMF = SMF + I2*FR(I)
1000      CONTINUE

          SUM = (SPE + SPF) + (SME + SMF)

          QMIE = 2.0*SUM/(XI*XI)

          RETURN

          END
```

```
      FUNCTION QDEI(RHO,G,TG)

C THIS IS THE DEIRMENDJIAN ANOMALOUS DIFFRACTION EXPRESSION

      QDEI = 2.0
      IF(RHO.GT.40.) RETURN

      EXG = EXP( - RHO*TG)
      CSG = COS(G)
      SRG = SIN(RHO - G)
      CRG = COS(RHO - 2.0*G)
      C2G = COS(2.0*G)

      QDEI = 2.0 - 4.0/RHO*CSG*(EXG*SRG - CSG/RHO*(C2G
     * - EXG*CRG))

      RETURN

      END


      FUNCTION QAD(RHO)

C THIS IS THE PURE ANOMALOUS DIFFRACTION EXPRESSION, WHICH
C ASSUMES NO ABSOPRTION.

      QAD = 2.0
      IF(RHO.GT.40.) RETURN

      QAD = 2.0 - 4.0/RHO*(SIN(RHO) - (1.0 - COS(RHO)) /RHO)

      RETURN

      END


      FUNCTION QRA(XI)

C THIS IS THE PURE RAYLEIGH LIMIT BLENDED INTO THE ABSORPTION
C LIMIT OF Qext = 2.0.

      DATA B,X0/2.6667,1.5/

      IF(XI.GT.X0) GO TO 100

      QRA = (XI/X0)**4
      RETURN

 100  QRA = 1.0 + TANH(B*(XI - X0))
      RETURN

      END
```

Opacity Model Listing

```
      FUNCTION DFACT(RHO,MR,TG)

C THIS IS THE DEIRMENDJIAN ADJUSTMENT FACTOR

      REAL MR,MR1

      FG = 2.0 + 4.0*TG*(1.0 + 0.75*TG)
      MR1 = MR - 1.0

      IF(RHO.LE.(5.0*MR1)) GO TO 100
      IF(RHO.LE.(4.08/(1.0 + 3.0*TG))) GO TO 300
      IF(RHO.LE.(4.08/(1.0 + TG))) GO TO 500

      DFACT = 1.0 + 2.04*MR1*FG/(MR*RHO*(FG - 1.0))
      RETURN

100   DFACT = 1.0 + MR1*MR1/(1.632*MR)*FG + (0.2*RHO -
     * MR1)/(MR1*FG)
      RETURN

300   DFACT = 1.0 + MR1*FG*RHO/(8.16*MR)
      RETURN

500   DFACT = 1.0 + MR1*FG/(2.0*MR*(1.0 + 3.0*TG))
      RETURN

      END
      FUNCTION FRACT(DG,SG,DP)

C   SUBROUTINE TO CALCULATE FRACTION OF PARTICLES AT SIZE DP

      DATA SQTWPI/2.506628/

      D0 = EXP(ALOG(0.5*DG) - 3.0*(ALOG(SG))**2)
      T2 = ALOG(0.5*DP/D0)
      T3 = ALOG(SG)
      T4 = AMIN1(ABS(T2/T3),6.0)
      FRACT = EXP( - 0.5*T4*T4)/(SQTWPI*T3*DP)

      RETURN

      END


      SUBROUTINE TEST(VL,VU,VW)

C THIS ROUTINE TESTS VL FOR BEING BETWEEN VW AND VU.  IF NOT,
C IT IS ASSIGNED THE CLOSER VALUE.

      IF (VL.LE.VU) GO TO 100
      WRITE(1,1) VL,VU
  1   FORMAT('+'//'+The value'F10.2,' is too big. ',F10.2,
     * ' will be used.')

      VL = VU

      CALL DELAY

      RETURN

100   IF (VL.GE.VW) RETURN
      WRITE(1,2) VL,VW
  2   FORMAT('+'//' + The value ,'F10.2,'  is too small. ',F10.2,
     * ' will be used.')

      VL = VW

      CALL DELAY

      RETURN

      END

      SUBROUTINE DELAY

      DO 200 J = 1,5
      DO 200 I = 1,10000
200   CONTINUE

      RETURN

      END
```

```
        SUBROUTINE QCAL(N)
 C PROGRESS OF CALCULATION
        LOGICAL X
        DATA IADR/X'3E90'/IGRF/140/
        DATA X/X'A0'/

        DO 200 I = 1,N
        IVDR = IADR + I

        CALL POKE(IVDR,X)

  200   CONTINUE

        RETURN

        END


        SUBROUTINE PLOT(N)
 C THIS ROUTINE PUTS A GRAPHIC SPOT ON LINE 10 OF THE SCREEN
        DATA IADR/X'3ED0'/IGRF/140/

        IVDR = IADR + N

        CALL POKE(IVDR,IGRF)

        RETURN

        END
```

Appendix A

Adaptations for Use with IBM-PC

APPENDIX A. ADAPTIONS FOR USE WITH IBM-PC

The following programs have been adapted for use with the IBM-PC:

ESP Section Failure Model
In-Stack Opacity Calculator
Plume Opacity Prediction Model
GCA/EPA Baghouse Model

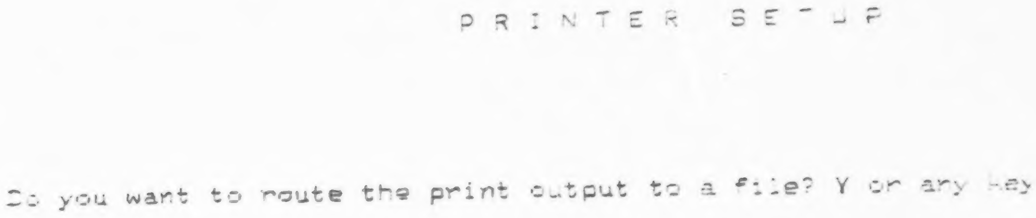
In general, those programs that have been adapted for use with the IBM-PC run in the same manner on both machines. This section outlines the differences that may be encountered. On the PC the <↵> key is same as the <ENTER> key. One addition that has been made to each of these programs is the option to send printer output to a file instead of the printer. When the program is initiated, a screen display will prompt the user for a choice between printer and file output (see Figure A-1).

Note that the required form for file names is different for the PC and the TRS-80. One example is shown in Figure A-2. These differences are reflected in the requests for file names. Follow the instructions given in the program or refer to the DOS manual.

Occasionally during operation, the user will encounter slight numerical differences between the two versions of these programs. These are due to the larger number of significant figures carried by the IBM-PC and should not be considered a problem. Some of these instances, as well as other changes, are noted below by the programs in which they occur.

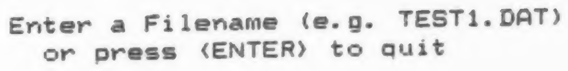
HARDWARE/SOFTWARE REQUIREMENTS

Every attempt has been made to make these programs hardware-independent so that they may be used on any microcomputer running version 2.0 (or higher) of the MS-DOS (Microsoft Corp.) operating system. All of these programs have been successfully run on an IBM-PC, IBM PC-XT, and Tandy 2000. Minimum system requirements include 256K of RAM, 1 floppy diskette drive, a printer, and the MS-DOS operating system. The programs do not make use of color graphics, so there are minimal video requirements.



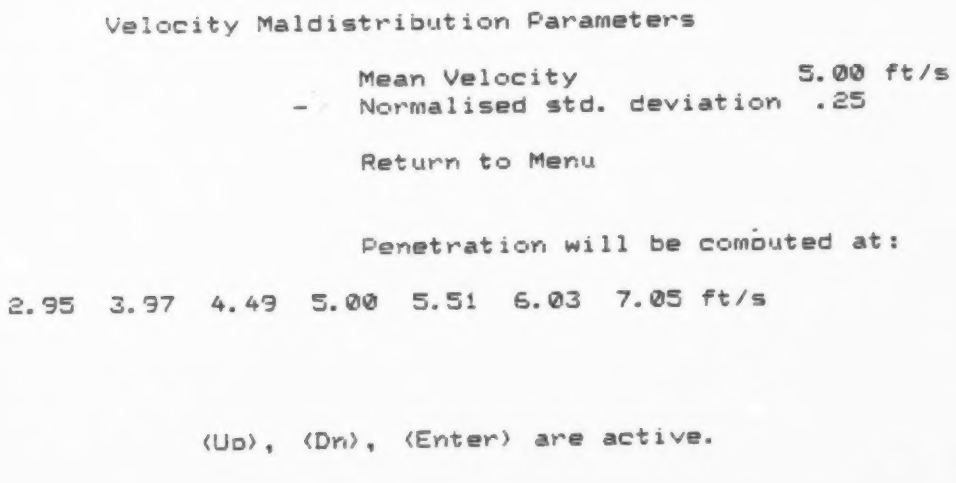
```
                P R I N T E R   S E T U P



Do you want to route the print output to a file? Y or any key
```

Figure A-1. Output options.



```
Enter a Filename (e.g. TEST1.DAT)
  or press <ENTER> to quit
```

Figure A-2. Request for file name.



```
     Velocity Maldistribution Parameters

                  Mean Velocity              5.00 ft/s
               -  Normalised std. deviation  .25

                  Return to Menu


                  Penetration will be computed at:

2.95  3.97  4.49  5.00  5.51  6.03  7.05 ft/s



           <Up>,  <Dn>,  <Enter> are active.
```

Figure A-3. IBM-PC velocity maldistribution parameters for the sample problem.

All of the programs that have been adapted for use with MS-DOS are written in Microsoft FORTRAN (V3.2). There are also some assembly language routines that were compiled with the MASM program. The FORTRAN compiler will be necessary if the user wishes to modify the source code. There should be no need to modify the assembly language routines.

ESP SECTIONAL MODEL

For the printout to appear as in the sample figures, the printer must be set up to print a 132-column line.

In the sample problem, when the Velocity Maldistribution Parameters appear on the screen (Figure 2-10 of Chapter 2) one of the numbers disagrees slightly as in Figure A-3.

IN-STACK OPACITY CALCULATOR

When entering data in this version, press <ENTER> instead of <D> in order to use the default value.

Another difference occurs during the calculations. Since the PC is much faster, the intermediate results will be flashed only for the Full Mie option.

Be sure to use capital "R" and "H" when attempting to return from a subroutine or seek help.

PLUME OPACITY PREDICTION MODEL

This program runs the same on both microcomputers except for the difference in allowable file names.

GCA/EPA BAGHOUSE MODEL

This program runs the same on both microcomputers except for the difference in allowable file names.

Appendix B

The Importance of Loss Mechanisms in Modeling Electrostatic Precipitator Performance

THE IMPORTANCE OF LOSS MECHANISMS IN MODELING ELECTROSTATIC PRECIPITATOR PERFORMANCE

By: Phil A. Lawless
Research Triangle Institute
P. O. Box 12194
Research Triangle Park, North Carolina 27709

Leslie E. Sparks
Industrial Environmental Protection Agency
U. S. Environmental Protection Agency
Research Triangle Park, North Carolina 27711

ABSTRACT

Although it has long been recognized that the precipitation rate parameter for high efficiency precipitators cannot be readily calculated from models of the collection process, the cause of the problem has not been clear. By examining the role and relative importance of various loss mechanisms, a simple model has been developed which explains the discrepancy between measured performance of precipitators and predictions of their collection performance. The model identifies section sneakage and rapping losses as critical to the performance of a precipitator and shows in which sections the collection performance is so dominated by losses that electrical improvements are of little value. Empirical and theoretical considerations of the loss mechanisms will be discussed.

This paper has been reviewed in accordance with the U. S. Environmental Protection Agency's peer and administrative review policies and approved for presentation and publication.

INTRODUCTION

The usual way of describing the collection performance of an electrostatic precipitator is by the Deutsch-Anderson equation(1). This equation relates the gas volume flow, the collection area, and the migration velocity of particles toward the collecting surfaces to the total efficiency of the unit. The form of the equation is:

$$\eta = 1 - \exp(- w\ A/Q), \qquad (1)$$

where η is the collection efficiency (expressed as a fraction), w is the migration velocity, A is the total plate area, and Q is the volume flow rate.

This equation can be derived for a single particle diameter in several ways(2), and has been successful in describing laboratory precipitators fairly well. But in full-scale precipitators, so many factors of polydispersity, fractional collection, back corona, and other nonideal conditions are encountered that the migration velocity can no longer be accurately calculated, and it is often referred to as an empirical "precipitation rate parameter."

Since the quantity A/Q also appears frequently in reporting precipitator performance, it is given the name "specific collection area" (SCA); and it is convenient to use the term "penetration" to characterize precipitator performance, with penetration, P, defined by:

$$P = 1 - \eta. \qquad (2)$$

Table 1 enumerates the performances of several high efficiency precipitators, with measured SCAs and calculated precipitation rate parameters. The rate parameters are plotted against SCA in Figure 1, and although there is much scatter in the results, there is a definite trend for w to decrease with increasing SCA.

Such behavior is not easily explained on the basis of the principles of particle transport in precipitators, but it can be explained phenomenologically if there is a relatively steady background of emissions that is independent of the collection of particles. A background level, if it greatly exceeded the collection-related emissions, would lead to "loss-limited" performance in which the product of precipitation rate and SCA is constant. Combining Equations 1 and 2 and taking logarithms of both sides, we obtain:

$$\ln(P) = - w \cdot SCA. \qquad (3)$$

For the units in Table 1 (excluding 9 and 14), the values of -ln(P) range from 4.6 to 7.2 and are relatively independent of SCA, thus confirming that w should vary inversely with SCA.

Loss-limited performance can be expressed in another way: if losses truly dominated collection, then there would be no improvement in collection with increases in SCA. As it is, the data from Table 1 plotted in Figure 2 indicate that the logarithm of the penetration decreases roughly as $SCA^{1/6}$,

261

TABLE 1. PERFORMANCE OF SEVERAL HIGH EFFICIENCY PRECIPITATORS

Unit	Measured penetration	SCA (s/m)	w (cm/s)
1. EPRI 1* (Naughton)	0.00078	147	6.42
2. EPRI 2* (Colbert)	0.0045	49	11.1
3. EPRI 2* (reduced voltage)	0.0097	49	9.48
4. EPRI 3* (Johnsonville)	0.00105	50	13.6
5. EPRI 4* (Allen)	0.0066	68	7.38
6. Allen †	0.0078	78	6.22
7. EPRI 5* (Jim Bridger)	0.0015	120	5.42
8. EPRI 6* (Navajo)	0.0098	55	8.38
9. EPRI 6* (half load)	0.0783	116	2.22
10. George Neal No. 3‡	0.003	147	3.95
11. George Neal No. 3‡ (last section off)	0.0103	110	4.15
12. George Neal No. 3‡ (partial load)	0.0011	250	2.72
13. George Neal No. 2§	0.0089	36	13.2
14. Wet ESP #	0.038	53	6.16

* Reference 3.

† Reference 4

‡ Reference 5.

§ Reference 6.

# Reference 7.

which is a very slowly varying function. Thus, loss mechanisms apparently do play a significant role in determining precipitator performance and should be adequately accounted for in an ESP model.

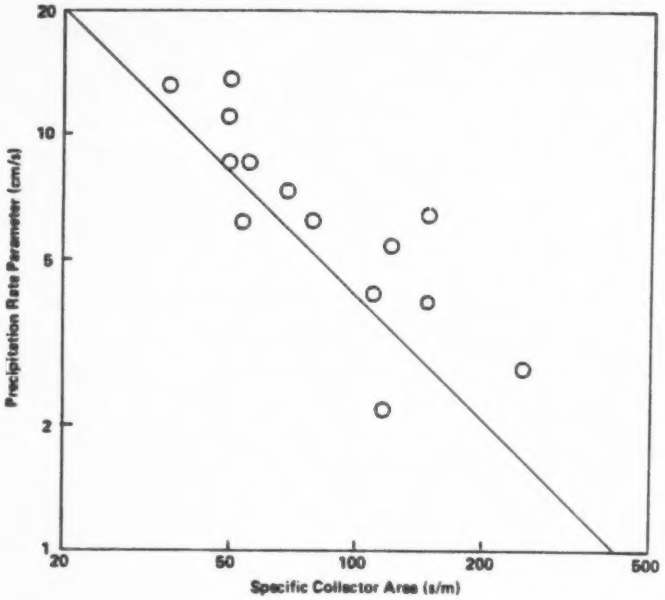


Figure 1. Precipitation rate parameter for several modern precipitators.

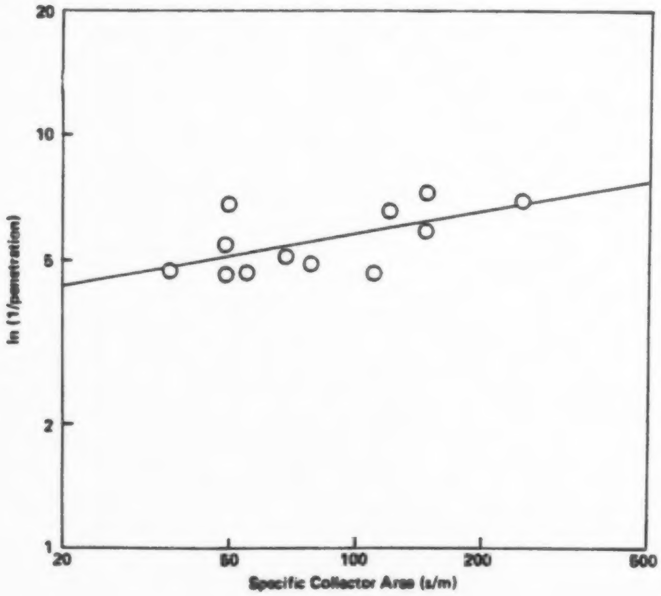


Figure 2. Variation of penetration with SCA for precipitators in Table 1.

SECTIONAL MODEL

Since modern precipitator design uses a high degree of sectionalization, it is appropriate to use a model that is similarly sectionalized. The main loss mechanisms that should then be accounted for are section bypassing (called sneakage) and reentrainment of collected material. These particular losses have been recognized as serious, and steps have been taken to reduce them(1) and measure them in operating units (3).

Figure 3 shows schematically how these losses are related in the precipitator section. Sneakage is assumed to totally bypass the collection zone in this model In so doing, sneakage effectively increases the SCA of the section since it lowers the flow through the active collection region. Reentrainment is that fraction of collected material which reenters the main flow. It is assumed to be a one-step process, even though in reality material may be reentrained and recollected many times in the same section.

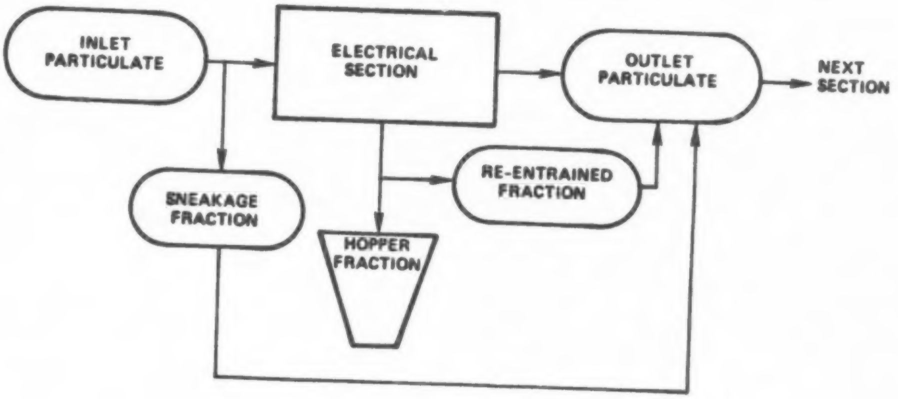


Figure 3. Operating model of a precipitator section.

The sectional model assumes that the overall precipitator penetration is the product of all the section penetrations. Some discussion of this assumption is made in a later section, but in the first approximation, it is accurate.

ACCOUNTING FOR SNEAKAGE

To begin constructing the model, we first assume reentrainment is negligible. If we designate the section penetration as P_n, the penetration due to sneakage as SN, and the penetration due to imperfect collection as P_c, then:

$$P_n = SN + (1 - SN) P_c. \tag{4}$$

As noted before, P_c may depend implicitly on SN. This relation shows that in the limit of very low P_c (good collection performance), the limit of penetration is set by the sneakage fraction.

ACCOUNTING FOR REENTRAINMENT

If we designate the reentrained fraction RR (for rapping reentrainment), then RR times the amount of material collected within a section is the amount reentrained. The resulting section penetration is:

$$P_n = SN + (1 - SN)\,P_c + RR\,(1 - SN)(1 - P_c). \qquad (5)$$

This equation shows that, in high efficiency sections, the relative roles of sneakage and reentrainment are comparable: if $P_c = 0$, then:

$$P_n = SN + RR\,(1 - SN). \qquad (6)$$

Equation 5 can be conveniently regrouped if we define a loss factor, LF, as:

$$LF = SN + RR - SN \cdot RR. \qquad (7)$$

Then Equation 5 may be written:

$$P_n = LF + (1 - LF)\,P_c, \qquad (8)$$

a form that is analogous to the sneakage equation, Equation 4. It is seen that the total loss factor, LF, sets the lower limit on section penetration even if the collection process is perfect.

This accounting for reentrainment differs from that of Reference 3. In Reference 3, it was assumed that all of the mass appearing at the outlet of the precipitator due to rapping was from the last section. This model assumes that each section contributes reentrained material to the succeeding sections and, ultimately, to the outlet. In fact, because of the sneakage around each section, rapping emissions from the earlier sections can appear directly in the outlet, although at reduced levels.

This direct contribution has been identified on at least two precipitators(8,12); in Reference 8, knowledge of the rapper program allowed direct correlation with opacity spikes. Rapping spikes from individual sections may not be observed on large ESPs because of the practice of subdividing the rapping of a section to avoid momentarily high emissions.

MODEL PREDICTIONS

To verify the accuracy of the model, it is necessary to define the loss factor and calculate the collection penetration function. These parts of the model involve several assumptions, which are kept as simple as possible.

First, the value of SN is assumed to be 0.1. This value for SN can be obtained by comparing the hopper and busbar volumes of an ESP with the total precipitator volume. It is not a very precise estimate, but the compensating effect on the SCA makes an exact value less critical.

A value of RR can be obtained from the measurements of Reference 3. There, the measured rapping emissions were fitted to the calculated mass on the plates of the last section by a power law:

$$Y = 0.155\ X^{0.905}, \tag{9}$$

where Y is the mass reentrained and X is the mass collected. The same data can be equally well fitted by the equation:

$$Y = 0.107\ X, \tag{10}$$

from which an experimental value of RR = .107 emerges. For convenience, this is rounded to 0.1.

In another reference, 9, load cell measurements of the change of mass of a precipitator inlet field with rapping amounted to 7 percent. Presumably, part of the mass rapped off would be caught in the hopper. This measurement places an upper limit on RR of about 0.05 but may apply only at the inlet field.

The collection penetrations are assumed to follow the Deutsch-Anderson model, with an appropriate choice of particle size. The SCAs of each electrical section are used, with the appropriate correction for sneakage. The migration velocity w is assumed to be the drift velocity of the chosen particle size, where the drift velocity, v, is given by:

$$v = q\ E/3\pi\ \mu\ d_p, \tag{11}$$

where q is the particle charge; E, the electric field; μ, the gas viscosity; and d_p, the particle diameter. It is also assumed that the particle diameter is large enough not to require the Cunningham correction.

The particle charge is assumed to be adequately described by field charging to saturation(1):

$$q = 3\pi\ \varepsilon_o\ d_p{}^2\ E, \tag{12}$$

where ε_o is the permittivity of free space, and other terms are as before. The charging and collecting electric fields are assumed equal so that combining Equations 11 and 12 gives:

$$v = d_p\ \varepsilon_o\ E^2/\mu. \tag{13}$$

The average field is reasonably well approximated by dividing the wire voltage by the wire-to-plate separation, with a correction factor:

$$E = V/1.25\ S_x, \tag{14}$$

where V is the operating voltage, and S_x, the wire-to-plate distance. More accurate estimates of E can be used, but this level of approximation is adequate for checking the model.

Back corona is encountered in some of the test cases and is responsible for two effects: back corona reduces the operating voltage of the ESP, and it reduces the charge on the particles. The first effect is automatically accounted for by using the measured voltages from the operational tests. The second effect is compensated for by multiplying the charge (Equation 12) by a factor of 0.3 or 0.5, according to whether the back corona is judged to be severe or moderate, respectively.

These multiplying factors are taken from measurements(10) which indicate that the severity of back corona is limited to a charge reduction of this order. The choice of multiplier is based upon the measured resistivities and shapes of the V-I curves in References 3 through 7.

The particle size, d_p, is chosen to be the mass median diameter (MMD) of all the particles entering the section. There are two main reasons for doing so: by using the MMD, half the total particulate mass (at sizes larger than the MMD) will be collected more effectively than the calculations indicate; and the MMD is usually large enough that Equations 11 and 12 are valid approximations.

In order for several sections to be placed in series, as in a real ESP, the MMD of particles emerging from a section must be calculated. Again, there are assumptions to be noted. It is assumed that the outlet MMD is a weighted average of the MMDs of particles penetrating due to sneakage, reentrainment, and imperfect collection. Then the outlet MMD can be calculated:

$$MMD_O = \frac{SN \cdot MMD_I + RR \cdot MMD_R + P_c\ MMD_P}{SN + RR + P_c}, \tag{15}$$

where MMD_O is the outlet MMD; MMD_I is the section inlet MMD; MMD_R is the characteristic fly ash rapping MMD, about 6 μm(3); and MMD_P is the MMD of the uncollected fly ash. A somewhat more accurate weighting would use Equation 5, but the improvement would not be noticeable for the cases considered.

The value of MMD_P depends on the collection performance of the section. If the section collects nothing, then MMD_P is the same as MMD_I. If the section collects very well, then MMD_P is usually in the range of 1 to 3 μm. Consequently, a linear combination of these values is used to determine MMD_P:

$$MMD_P = MMD_I \cdot P_c + 2 \cdot (1 - P_c), \tag{16}$$

where all the diameters are in micrometers.

COMPARISON WITH MEASUREMENTS

Using all these assumptions, with additional data furnished in References 3 through 7, model predictions of the penetrations of the ESPs in Table 1 were calculated. The resulting penetrations are shown in Table 2 and compared with the measurements in both Table 2 and Figure 4. It can be seen that the predicted values fall within about a factor of two of the measured values and that the prediction is, on the average, very close to the measured values.

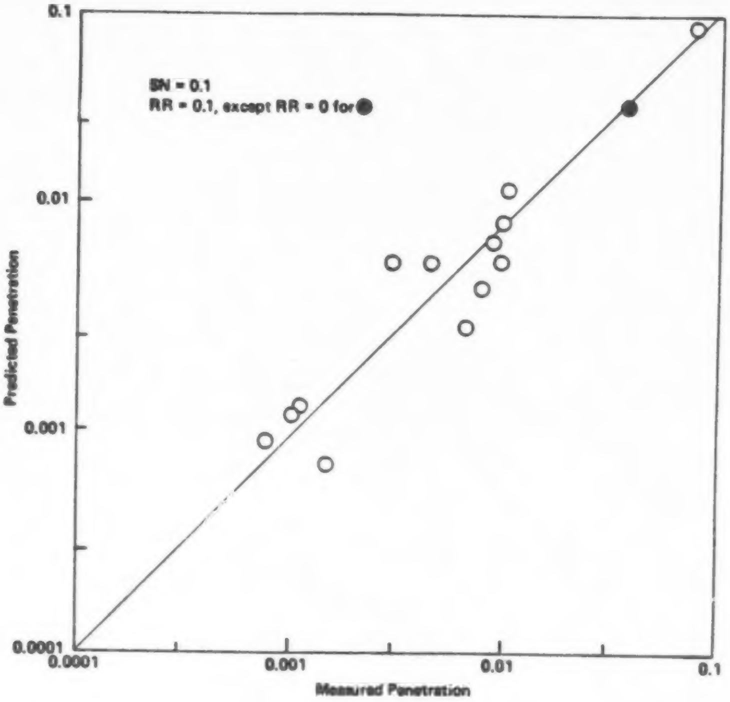


Figure 4. Comparison of model predictions with measured penetrations.

These results suggest that the chosen values of RR and SN are universally acceptable for similar ESPs, at least to the accuracy shown here.

EXAMINATION OF RESULTS

Inspection of the section penetrations for the 14 cases shows that several are completely loss limited; that is, the penetration of each section is very close to LF, as computed by Equation 8. These are Cases 2, 3, 4, 11, 12, and 13 from Table 1. If the experimental loss factors are computed according to:

$$LF_m = (P_m)^{1/N}, \qquad (17)$$

TABLE 2. COMPARISON OF MEASURED AND PREDICTED PENETRATION

Unit (Table 1)	P_m (measured)	P_c (calculated)	P_c/P_m
1	0.00078	0.00094	1.21
2	0.0045	0.0068	1.51
3	0.0097	0.0068	0.70
4	0.00105	0.0013	1.24
5	0.0066	0.0033	0.50
6	0.0078	0.0050	0.64
7	0.0015	0.00075	0.50
8	0.0098	0.0101	1.03
9	0.0783	0.0854	1.09
10	0.003	0.0066	2.20
11	0.0103	0.0142	1.38
12	0.0011	0.0014	1.24
13	0.0089	0.0082	0.92
14*	0.038	0.036	0.94
			Avg P_c/P_m = 1.08(±.46)

* RR = 0.

where P_m is the measured penetration and N, the number of sections; then the average value of LF_m is 0.193 ± 0.02 (one standard deviation limits). The computed value of LF (Equation 7) is 0.19; this is further support for the universal use of the chosen values for RR and SN.

There are cases in which several sections on the inlet side are loss limited but which are collection limited toward the outlet side: Cases 5, 6, 7, and 10. In these units, the change in behavior occurs because the MMD decreases from section to section to the point that the collection suffers. Finally, these are cases in which all sections show penetrations well above the loss limit: Cases 1, 8, 9, and 14. In the first three of these cases, the precipitators have 6 sections each; this would be valuable in limiting the effects of losses, but the sectional SCAs are not adequate to collect the particulate matter. In these cases, the high degree of sectionalization does not help the performance very much.

For those sections in which loss-limited performance is achieved, the penetration is insensitive to changes in voltage, SCA, or particle size. Loss-limited sections have a margin for good operation and can tolerate changes from their nominal operating points. Conversely, performance-limited sections will be responsive to all those parameters.

This operating margin for loss-limited ESPs offers a simple explanation for two puzzling precipitator phenomena: the variation of precipitation rate parameter with gas velocity and the wide ESP effect. The first phenomenon was noted in laboratory and pilot ESPs where it was found that the rate parameter w was an increasing function of gas velocity(1). This can be explained as a manifestation of loss-limited operation: because the SCA is an inverse function of the gas velocity and the penetration is constant, Equation 3 indicates the rate parameter should be proportional to velocity. The experience with wide spacing ESPs indicates they are capable of achieving the same overall performance as conventional ESPs, even though their SCAs are significantly less.(11) Again, this can be explained as a penetration determined by losses rather than by collection performance. (It seems reasonable that the operating margin of a wide ESP would be less than that of a conventional unit, all other factors being equal.)

THEORETICAL CONSIDERATIONS OF SNEAKAGE AND RAPPING REENTRAINMENT

With the importance of the loss mechanisms identified, a more detailed look at them is worthwhile. It seems unrealistic to think that a sneakage flow of 10 percent of the total could be completely mixed with the cleaned gas in the short distance between electrical sections. Yet, this is an implied assumption of the sectional model. It also seems unrealistic to think that rapping reentrainment is independent of the particle size distribution being collected, but this is an explicit assumption in this model.

Precipitators were sectionalized and baffled in the past when it was realized that direct bypassage of the precipitator would seriously degrade the performance. Consequently, direct sneakage paths are rare (except when part of a section is deenergized), and it is more likely that partially mixed paths are the prevalent mode for sneakage. A brief investigation of partial mixing models has indicated that rather small flows could have the same effect as a fully mixed sneakage of 0.1, but that the effect depends on the number of sections as well as the degree of partial mixing.

To illustrate, assume that there is a direct flow path along the precipitator that carries 1 percent of the inlet mass directly to the outlet without mixing. If there were no other sneakage paths, one section would have a sneakage 10 times lower than a similar conventional section with SN = 0.1, completely mixed. Two sections in series would have the same sneakage as two conventional sections. Three in series would have 10 times more sneakage than three conventional sections. If the direct path is approximately 50 percent mixed (or diluted) in each section, then three sections would have the same sneakage as three conventional sections. It is much more acceptable to conceive of 1-percent-sneakage flows with 50-percent mixing than the levels suggested in the model. The question needs further investigation.

Rapping reentrainment can be analyzed somewhat more completely. From Equation 5, the reentrained material penetration, designated P_r, is:

$$P_r = RR\ (1 - SN)(1 - P_c). \qquad (18)$$

If we make the assumption that the section collects reentrained material as well as it collects original material and that all the collected material is initially reentrained, then $RR = P_c$, and the reentrainment penetration becomes:

$$P_r = (1 - SN)\ P_c\ (1 - P_c). \qquad (19)$$

The function $P_c\ (1 - P_c)$ has a maximum value of 0.25 at $P_c = 0.5$, giving a maximum reentrainment penetration of 0.225 (for SN = 0.1). The actual collection penetration of most sections is less than 0.5 for most particle sizes, resulting in a lower overall rapping emission.

Under the same assumptions, a size-dependent reentrainment fraction, $\overline{RR}$, can be calculated from knowledge of the size distribution:

$$\overline{RR} = \int_0^\infty P_c\ (d_p)\ f\ (d_p)\ dd_p, \qquad (20)$$

where $f\ (d_p)$ is the distribution function of particle sizes of diamter d_p. Since each section modifies the size distribution for succeeding sections, it is expected that the average reentrainment coefficient would vary from section to section. Again, further investigations are necessary.

SUMMARY

It has been shown that the loss mechanisms of sneakage and rapping reentrainment play a very significant role in determining precipitator performance. Their dominant position requires proper accounting for their effects in any model, but with the proper structure even a simple model based on fundamental principles is capable of very good results.

The implications of loss-limited performance are significant for new precipitator designs and for upgrades to existing precipitators. It appears that many existing installations are loss limited and that several precipitator phenomena are manifestations of loss-limited performance.

With the loss mechanisms so important in the overall operation of precipitators, research resources ought to be directed to theoretical and experimental investigations of these processes, since major improvements in precipitator performance seem possible only in those areas.

REFERENCES

1. White, H. J. Industrial Electrostatic Precipitation. Addison-Wesley, Reading, Massachusetts, 1963.

2. Oglesby, S., Jr. and Nichols, G. B. Electrostatic Precipitation. Marcel Dekker, New York, New York, 1978.

3. Gooch, J. P. and Marchant, G. H. Electrostatic precipitator rapping reentrainment and computer model studies. Report No. FP-792. Electric Power Research Institute, Palo Alto, California, 1978.

4. Drehmel, D. C. and Gooding, C. H. Field test of a hot-side electrostatic precipitator. EPA-600/2-77-208. NTIS PB274-017/AS. U. S. Environmental Protection Agency, Research Triangle Park, North Carolina, 1978.

5. Ensor, D. S., Cowen, S. J., Hooper, R., and Markowski, G. Evaluation of the George Neal No. 3 electrostatic precipitator. Report No. FP-1145. Electric Power Research Institute, Palo Alto, California, 1979.

6. Dismukes, E. B., and Gooch, J. P. Fly ash conditioning with sulfur trioxide. EPA-600/2-77-242. NITS PB276-651/AS. U. S. Environmental Protection Agency, Research Triangle Park, North Carolina, 1977.

7. Gooch, J. P., McCain, J. D., and Sparks, L. E. Performance of a wet electrostatic precipitator in an aluminum processing facility. EPA-600/2-77-208. NTIS PB274-017/AS. U. S. Environmental Protection Agency, Research Triangle Park, North Carolina, 1978.

8. Ensor, D. S., Lawless, P. A., Damle, A. S., and Sparks, L. E. Evaluation of the United McGill electrostatic precipitator. EPA-600/9-82-005a. NTIS PB83-149-583. U. S. Environmental Protection Agency, Research Triangle Park, North Carolina, 1982.

9. Spencer, H. W. and Juricic, D. Electrostatic precipitator plate rapping and reliability. Report No. FP-1006, Volume 3, Part 2. Electric Power Research Institute, Palo Alto, California, 1980.

10. Masuda, S. and Nonogaki. Bi-ionized structure of back discharge field in an electrostatic precipitator. In: Conference Record IEEE-IAS Annual Meeting, 1981. Lib. Cong. No. 80-640527. p. 1111.

11. Heinrich, D. O. Electrostatic precipitator collector spacings above 300 mm (12")? Atmospheric Environment. 13; 1707, 1979.

12. Sproull, W. T. Minimizing rapping loss in precipitators at a 2000 megawatt coal-fired power station. JAPCA. 22: 181, 1972.

Appendix C

Model Studies of Electrostatic Precipitator Rapping Strategies

MODEL STUDIES OF ELECTROSTATIC PRECIPITATOR RAPPING STRATEGIES

By: Phil A. Lawless
Research Triangle Institute
P.O. Box 12194
Research Triangle Park, North Carolina 27709

Leslie E. Sparks
Industrial Environmental Research Laboratory
U. S. Environmental Protection Agency
Research Triangle Park, North Carolina 27711

ABSTRACT

A simple electrostatic precipitator model was used in a dynamic mode to examine the effects of rapping strategies on precipitator performance. The model allowed for rapping on a timed basis or on a layer thickness basis. The model was used to evaluate the mean mass emissions and mean opacity as well as the puffs due to rapping. A comparison of the results with common practices was made. The model confirms that rapping emissions contribute very heavily to the total precipitator emissions, that distributed rapping reduces the size of opacity spikes and demonstrates that inlet field rapping may contribute more to emissions than outlet field rapping.

This paper has been reviewed in accordance with the U. S. Environmental Protection Agency's peer and administrative review policies and approved for presentation and publication.

INTRODUCTION

Electrostatic precipitators have been modeled as static processes for many years. The Deutsch-Anderson model(1) is probably the best known static model of a precipitator (ESP), but it has been followed in recent years by models that attempt to account for dynamic rapping effects by appropriate average behavior(2,3). The problem with using static models to describe the essentially dynamic behavior of an ESP is that the average behavior may be modeled very well without understanding all the effects of the process.

As a result, although static models have demonstrated the importance of rapping as a factor in ESP emissions(2,3,4), they are inadequate to determine the magnitude of opacity spikes due to rapping, or to show how rapping strategies affect mass emission and opacity levels. A dynamic precipitator performance model can answer these problems and can, furthermore, offer a substantial compression of time scales. That is, effects which might take days to become evident on a real ESP can be modeled in a few minutes.

Therefore, a dynamic ESP model has been developed to investigate these effects. Although the model is fairly simple, it is based upon a successful static model(4,5) and operates according to established principles. The version described here runs on a microcomputer and is designed for interactive use.

MODEL STRUCTURE

The dynamic model is based upon the sectional model structure described in References 4 and 5. The sectional structure relates the contributions of sneakage, rapping reentrainment, and imperfect collection to the total emissions from a given electrical section. A schematic of the structure is shown in Figure 1.

The mathematical description of these contributions are written with the following definitions:

P_c - the section penetration due to imperfect collection;

SN - the fraction of the particle stream which bypasses the collection zone;

RR - the fraction of collected material which is reentrained by rapping; and

P_n - the total section penetration.

Then, assuming for the moment that there is no rapping contribution (RR = 0), the section penetration is:

$$P_n = SN + (1 - SN)\, P_c. \qquad (1)$$

This equation shows that sneakage sets a lower limit on the penetration of the section, in the absence of rapping.

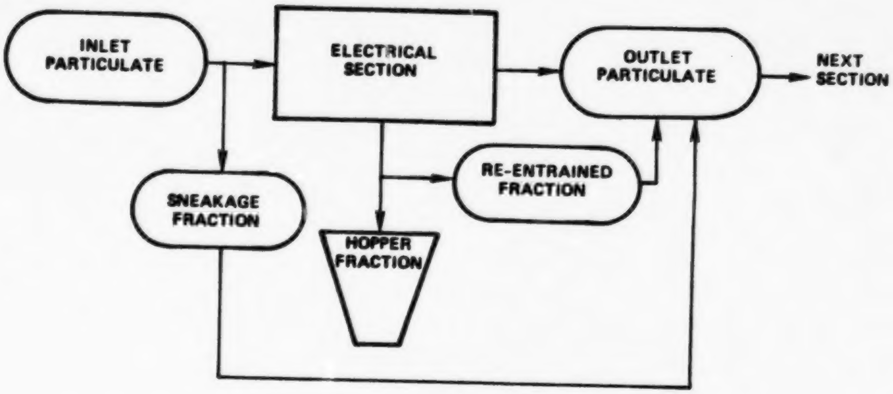


Figure 1. Sectional model of an ESP showing major loss mechanisms.

Now, since only previously collected material can be reentrained, the total section penetration in the presence of rapping is given by:

$$P_n = SN + (1 - SN)\ P_c + RR\ (1 - SN)(1 - P_c). \qquad (2)$$

At this point, the dynamic model diverges from the static model. In the static model, the total precipitator penetration is given as the product of all the sectional penetrations and is independent of time. Rapping emissions contribute to each section's penetration, as if the precipitator were in a state of continuous rapping. In the dynamic model, on the other hand, Equation 1 gives the section penetration, independent of time. During the period between raps, the particulate material is accumulated on the collection plates of each section, and, when a rapping event occurs, the fraction RR of the mass is added to the burden for subsequent collection and emission.

At a given time, the total material present on any plate is the sum of material caught by the steady collection process plus the accumulated reentrained material from upstream sections. The emission level at any time is the sum of the material penetrating all the way from the inlet of the precipitator plus any material penetrating from a rapped section.

In spite of these differences, the calculation of the section penetration is treated similarly to the static model penetration. That is, the section collection performance is assumed to obey the Deutsch-Anderson equation:

$$P_c = \exp\,(-\ w \cdot SCA), \qquad (3)$$

where w is the migration velocity of a particle and SCA (specific collector area) is the ratio of collection plate area to gas volume flow.

The migration velocity, w, is calculated from Stokes' law:

$$w = qE/3\pi\ \mu\ d_p, \qquad (4)$$

where q is the particle charge; E, the electric field; μ, the gas viscosity; and d_p, the particle diameter. By assuming the particle is large enough for field charging(1) to be valid, the particle charge is given by:

$$q = 3\pi\ \varepsilon_o\ E\ d_p^{\ 2}, \qquad (5)$$

where ε_o is the free space permittivity, and the other symbols are as before. Equations 4 and 5 can be conveniently combined to give:

$$w = \varepsilon_o\ E^2\ d_p/\mu. \qquad (6)$$

The remaining problem is to decide what particle diameter is appropriate to use. For reasons discussed in Reference 4, the mass median diameter (MMD) of particles entering the section is used. All the rapped material is assumed to have an MMD of 6 µm(6), and the minimum MMD of the direct uncollected material is assumed to be 2 µm.

To compute the MMD of the material passing from one section to another, a weighted average of all the particulate sources is used:

$$MMD_o = \frac{SN \cdot MMD_I + (1 - SN)\ P_c \cdot MMD_c + M_R \cdot MMD_R}{SN + (1 - SN)\ P_c + M_R}, \qquad (7)$$

where MMD_o is the outlet MMD; MMD_I, the inlet MMD; MMD_c, the uncollected material MMD; MMD_R, the rapping MMD; and M_R, the mass of reentrained material. Because M_R is zero except during rapping, it can be seen that MMD_o may swing substantially during a whole rapping cycle. In addition, MMD_c depends on the actual collection performance of the section:

$$MMD_c = MMD_I \cdot P_c + 2 \cdot (1 - P_c), \qquad (8)$$

where all diameters are in micrometers. The only significance of Equation 8 is that a section with $P_c = 1$ should deliver the inlet MMD and should deliver progressively smaller values as more of the large particles are collected.

DYNAMIC EFFECTS

To make the dynamic model useful, it must provide outputs of quantities that relate to general experience, as well as quantities that reveal the behavior of the process. The observable outputs, corresponding to measurable quantities, are mass emissions and stack opacity values. The quantities that are not directly observable are the thicknesses of the dust layers in each section.

Normally, precipitator mass emissions are determined over a period of hours, averaging over many rapping cycles. Opacity, on the other hand, is measured on the basis of a 6-min running average, although some transmissometers can give instantaneous (unaveraged) opacities. Therefore, the model makes available instantaneous values of outlet dust concentration and opacity, a 6-min running average of opacity, and a running average of outlet concentration, with averaging periods of up to 4 hrs. In this way, the user can correlate instantaneous values with typical measured quantities.

Real precipitators usually have the rapping in a given section distributed over several subsections. This is done to avoid large excess dust concentrations in the outlet. The model has incorporated this feature, allowing for the successive rapping of subsections after the initiation of a given section rapping cycle.

There are normal process variations in the dust sources which precipitators control; the model allows for setting the average concentration of dust delivered from the source and for adding a random deviation around the average value. The resulting variations can be used to determine how sensitive the outlet emissions are to process changes and how long an averaging time is required to reduce them.

The last areas of dynamic effects, which the present version does not account for, are particle charging and dust cake mechanics. The model assumes that particle charging is instantaneous, but there are good grounds to think that rapping loads the gas stream so heavily with dust that charging may take a substantial time. In at least one experimental measurement(7), the rapping puffs were found to emerge from a precipitator with a polarity opposite the normal value. The model also neglects any changes in dust cake characteristics with accumulated thickness. Such changes might affect the amount reentrained with each rap, but there is little information on how much change could be expected.

COMPARISON WITH THE STATIC MODEL

The static model and the dynamic model have the same governing equations if the dynamic model experiences a rap during each period of calculation. Therefore, it is expected that their predictions should also be nearly the same for somewhat less frequent rapping; this result is observed. At still longer rapping intervals, the two models diverge in their predictions, with the dynamic model predicting greater average mass emissions and large rapping spikes in opacity. The differences are not severe, but it is not clear whether the behavior is realistic or just an artifact of the model.

RAPPING STRATEGIES

There are three basically different rapping strategies that can be used in precipitator operation. We call them "short-cycle rapping," "staged-cycle rapping," and "thickness rapping."

SHORT-CYCLE RAPPING

Short-cycle rapping is a strategy in which there is nearly continuous rapping of some part of the precipitator. All the sections use the same rapping period, and the length of the period is only a few minutes. With distributed rapping in each section, the effect is close to continuous rapping. This strategy is used on several high efficiency precipitators(6).

The purpose of short-cycle rapping is to avoid rapping puffs, if at all possible; it also achieves a rapping condition which the dynamic model predicts gives the lowest emission rate. A possible disadvantage of this strategy is that very little layer buildup occurs in the outlet sections, and so rapping may disperse the dust more than dislodge it.

STAGED-CYCLE RAPPING

This strategy uses longer rapping periods on successive sections of the ESP. While the inlet section may be rapped as frequently as in the short-cycle mode, the outlet field may be rapped only once an hour, or longer(8).

The purpose of this strategy is to allow more dust to accumulate in the later sections, with perhaps less reentrainment during rapping. Also, the number of rapping puffs in a given time interval is reduced over the short-cycle mode. A possible disadvantage of this strategy is that the outlet sections will create larger rapping puffs.

THICKNESS RAPPING

In this mode, the dust layer in each section is allowed to build up to a predetermined thickness before rapping is initiated. The thickness might be allowed to vary from section to section, but in the model, a common thickness is used for all sections. No examples of this mode are known to be used.

The purpose of this strategy is to rap the sections only when an optimum thickness has been obtained. In this way, the graction reentrained in each section could be minimized. This mode would tend to give very large emissions from the last section, but at very infrequent intervals.

MODELING RESULTS

Since there are so many parameters that can be varied in a study like this one, a fairly standard configuration was chosen, and variations on the rapping cycle were tested. The basic configuration is summarized below:

Number of sections = 3
Total SCA = 29.5 s/m (150 ft^{2}2/kACFM)
Electric Fields = 2.5 kV/cm, each section
Temperature = 149° C (300° F)
Inlet MMD = 20 μm
Stack diameter = 6.1 m (20 ft)
Inlet loading = 4.6 g/m^{3} (2 gr/ACF)

The parameters that were varied included: rapper periods, critical layer thickness for rapping, distribution of rapping in one section, RR, and SN.

Using the static model(4), the calculated mass penetration was 0.012, for an efficiency of 98.8 percent. The corresponding stack opacity for the inlet loading was about 7 percent. The efficiency is very reasonable by accepted standards(1); the opacity is somewhat low for that level of emissions but not seriously so.

The penetrations of each section, using the static model, were 0.195, 0.218, and 0.275 for the first, second, and third sections, respectively. From the loss factor, 0.19, it can be seen that the first two sections are loss limited, and the third section is nearly loss limited(4).

SHORT-CYCLE RAPPING RESULTS

The rapper periods were set at 10 minutes for each section. The model introduces time offset for each section after the first so that no two sections may rap together. Thus, if the first section rap is counted as time zero, the second and third sections rap at 3 min and 6 min, respectively, into the cycle of Section 1.

The sections were also examined for the effects of subsectionalizing the rapping by removing one fifth of the collected dust in five successive raps. The average thickness of the dust remaining on the plates was calculated (assuming a 50 percent porosity) and compared in Figure 2. The average layer thickness is seen to vary considerably from one case to another, but implied in Figure 2 is the result that each rapping puff in the subsectioned case contains only one fifth the material as the single rap case.

The growth of the layers in all three sections is shown in Figure 3 for the single rap case. A major consequence of short-cycle rapping is that the average layer thickness decreases strongly in each section after the inlet, from one to another. But Figure 3 also indicates that the first section rapping contributes more material to the second and third sections than does the normal penetration of material. The first section rapping contribution even exceeds the second section's in the growth of the third section layer.

The instantaneous opacity, plotted in Figure 4, bears out the fact that the first section emissions dominate the outlet opacity, even though they must pass through all the remaining sections. The 6-min average opacity, also in Figure 4, tends to obscure the contribution of the first section. The highest values of the running average contain contributions from both the first and second sections together.

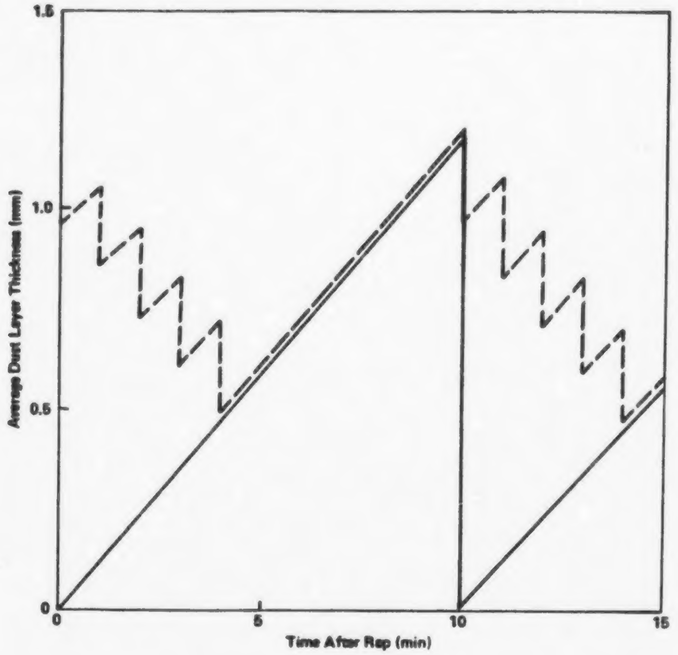


Figure 2. Effect of distributed rapping (---) on layer thickness.

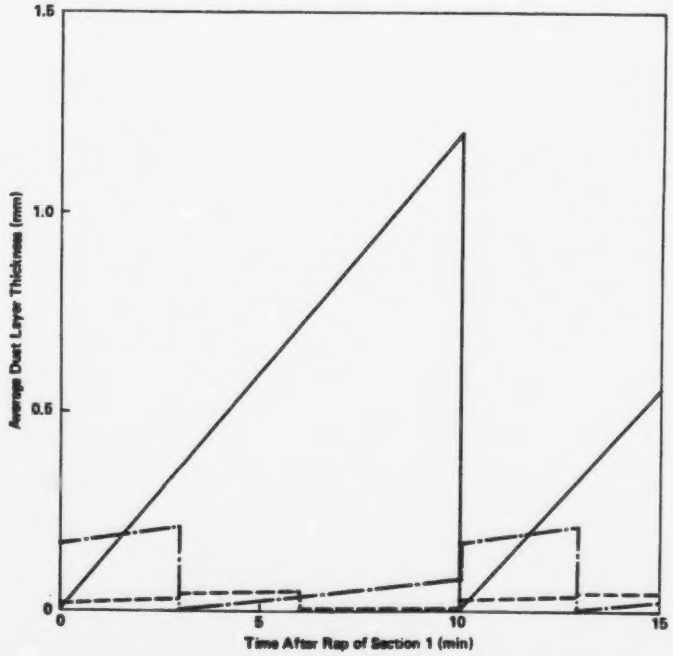


Figure 3. Layer thickness under short-cycle rapping for Section 1 (—), Section 2 (-·-), and Section 3 (---).

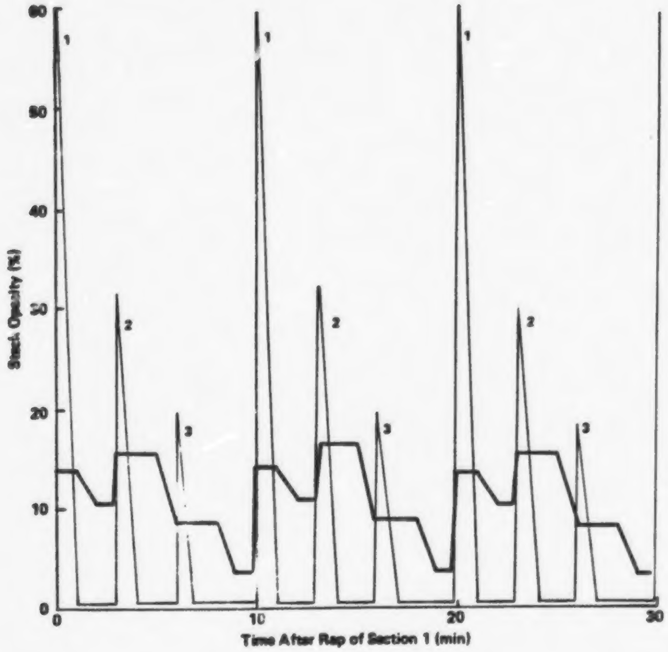


Figure 4. Opacity for short-cycle rapping: (—), instantaneous; (—), average.

The effect of subsectionalizing the rapping is to smooth the running average even more, keeping the opacity values between 7 and 11 percent. The instantaneous opacity values are reduced to a maximum of 17 percent, for combined first and second section peaks.

Short-cycle rapping, especially combined with distributed rapping, does serve its purpose of keeping the average opacity free of obvious opacity spikes. The model reveals the surprising result that the inlet section dominates in contributing to the instantaneous opacity and the average opacity as well (since all sections are rapped equally often). The contribution of rapping emissions to the total mass emissions are 44, 23, and 19 percent, respectively, for Sections 1, 2, and 3. The rapping emissions for all sections together account for 87 percent of the total.

STAGED-CYCLE RAPPING RESULTS

Using the same first section rapper period, 10 min, the rapper periods for the second and third sections were set for 20 and 40 min, respectively. The first section behaves just as the first section in the short-cycle cases. Similarly, the average layer thicknesses resemble those of Figure 3, except that the maximum thickness reached in Section 2 are twice, and those in Section 3 four times, the values reached in the short-cycle case.

The instantaneous and averaged opacities are shown in Figure 5, with a single rap per section. The instantaneous opacity spikes are nearly equal for all three sections for this particular choice of rapper periods. Because of the differences in frequencies of rapping, the inlet section still dominates in its contribution to the rapping emissions. The averaged opacity also exhibits larger variation than in the short-cycle case, both higher and lower. The occasional peaks are higher because more mass has been allowed to collect on the later sections, but the corresponding valleys are lower because of less frequent rapping.

It is a quirk of the model that distributed rapping does not improve the variation in average opacity. In several parts of the rapping cycles, there are overlapping periods with two sections rapping. This would be avoided on a real precipitator.

With regard to mass emissions, the rapping contributions are 51, 26, and 20 percent for Sections 1, 2, and 3. Rapping emissions, in this case, account for 97 percent of the total emissions. Nonetheless, the mass emission averages are nearly identical for the short-cycle and staged-cycle rapping.

The model indicates that staged-cycle rapping does give thicker dust layers in the later sections, but the opacity and mass contributions of the later stages relative to the first do decrease somewhat because of the less frequent rapping.

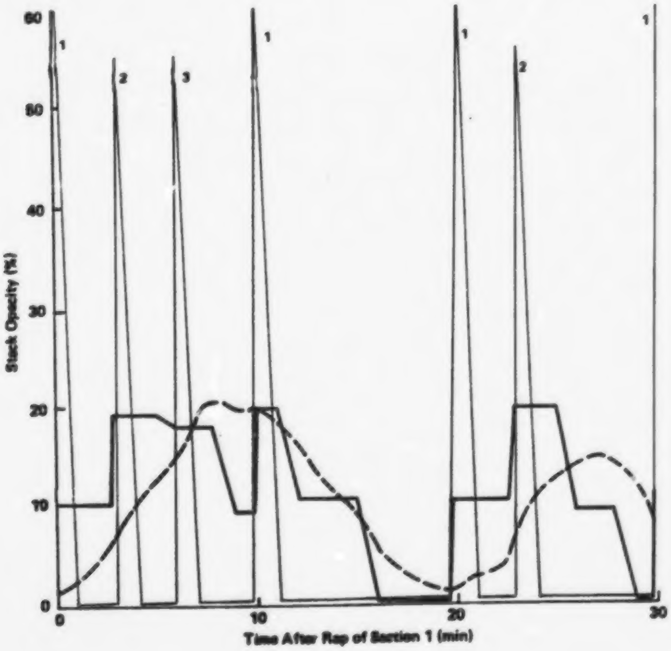


Figure 5. Opacity for staged-cycle rapping: (—), instantaneous; (━), average; (---), distributed.

THICKNESS RAPPING RESULTS

For thickness rapping, a critical value of thickness was set for all sections. When a given section thickness exceeded the critical value, a rapping cycle was initiated. Because of this method of initiating rapping, several sections often were rapped in rapid succession, and considerable overlap of rapping puffs occurred.

When the critical thickness was set small (less than 0.1 mm), the first section rapped continuously and still was unable to maintain the average layer thickness near the critical value. When the critical thickness was 2 mm, the third section rapped very infrequently.

Because the time scales are so different for the buildup of dust in each section, the length of time to reach a given thickness was plotted on a logarithmic scale in Figure 6. It is evident that much of the layer growth after Section 1 is due to rapping puffs from upstream sections. The first and second sections contribute almost equally to the growth in the third. The larger jumps in thickness are where both upstream sections rapped in sequence.

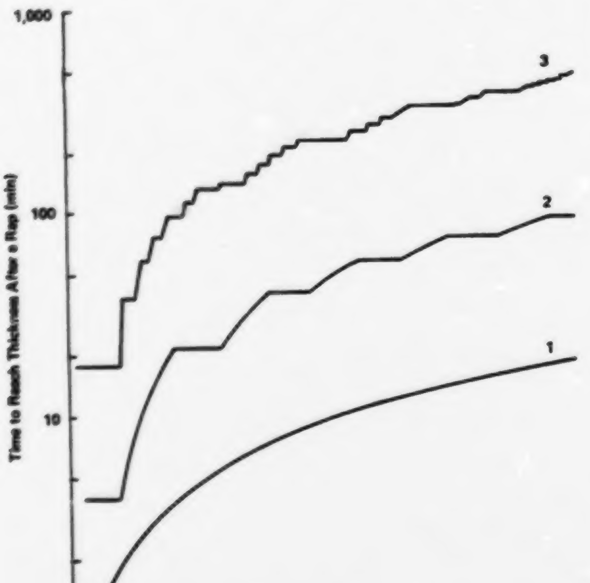


Figure 6. Buildup of layer depth for three sections under thickness rapping.

With a critical layer thickness of 2 mm, the instantaneous opacity spikes from all sections exceeded 90 percent. To show the relative amplitudes of the mass in each puff, a critical thickness of 0.2 mm was run, resulting in individual opacity spikes of less than 50 percent. A portion of this run is shown in Figure 7. The spikes from Section 1 are much lower in amplitude than in Figures 4 and 5 because of the higher frequency of rapping. The spikes from Section 2 greatly exceed those of Section 1; and, although not apparent in the figure, those of Section 3 exceed those of Section 2 substantially.

If this frequent rapping were supplemented by distributed rapping, Section 1 would be continuously rapped. Using distributed rapping, the maximum opacity spike was reduced to 24 percent, and the average opacity remained between 4 and 8 percent at all times.

Thickness rapping produced sectional rapping mass fractions of 34, 36, and 43 percent, with the rapping emissions contributing 96 percent of the total. (The section percentages sum to more than 100 percent because some contributions are counted twice with the overlap of rapping.)

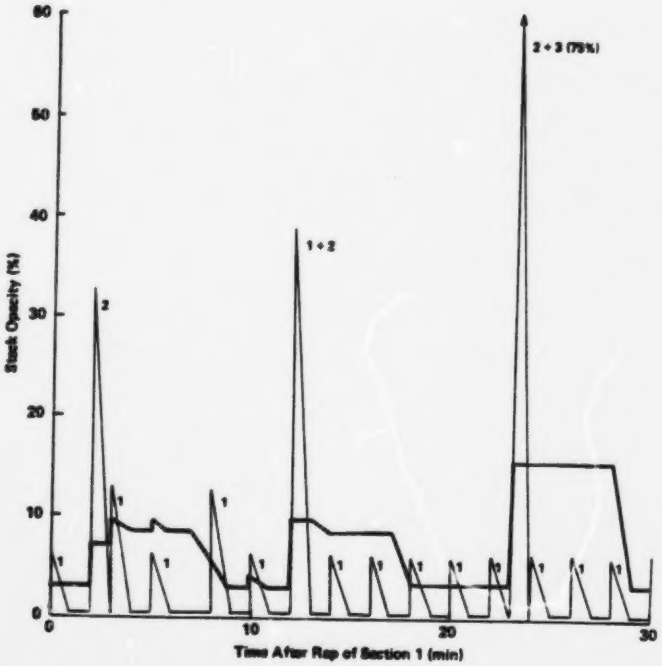


Figure 7. Opacity for thickness rapping: (—), instantaneous; (━), average.

Thickness rapping does achieve the purpose of equalizing the thickness accumulated in each section. It further equalizes the mass contributed by each section to the total emissions but at the cost of greater opacity spikes from the later sections. The actual amplitudes of the spikes are a function of the critical thickness chosen and decrease with smaller values.

SUMMARY OF RESULTS

From Figure 6, it can be seen that for this model precipitator, thickness rapping produced rapping periods from section to section in the ratio of 5 to 1. Since short-cycle rapping produces section-to-section ratios of 1 to 1, it is enlightening to examine effects on opacity of ratios between those extremes. This is shown in Figure 8 for the same precipitator, having a fixed first section rapping period of 6 min instead of 10 as before.

Although the opacity for Section 3 is near saturation at the high ratios, the opacity exhibits a linear trend at lower ratios. Since the rapper period ratio of 2 makes all the rapper spikes essentially equal in amplitude, there would appear to be a significant advantage for operating at those periods since no one section would cause excessive opacity spikes.

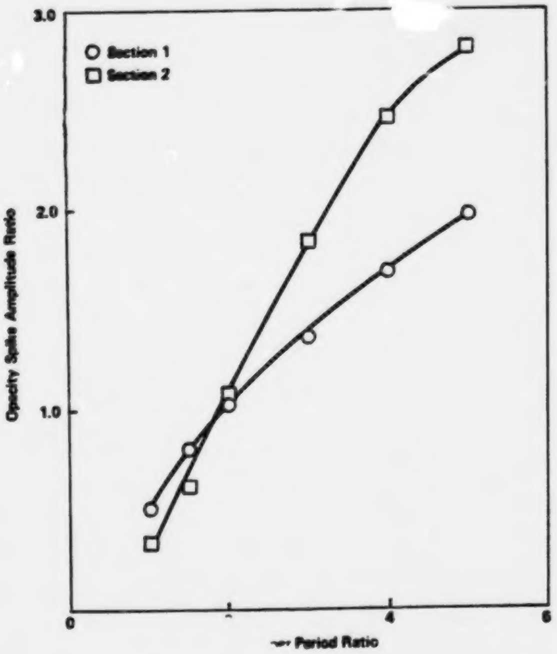


Figure 8. Opacity spike amplitudes referenced to Section 1 for various rapper periods.

Figure 8 indicates that the three rapping strategies studied here are part of one continuum. Each rapping strategy has its particular advantages and disadvantages. In all of them, rapping emissions account for almost all the mass emissions from the precipitator; and, except for the thickness rapping strategy, the first section contributes more than the succeeding sections.

The technique of distributing the rapping over many subsections does reduce the magnitude of opacity spikes substantially. But concurrent rapping of two or more sections can produce combined spikes of large amplitude, particularly if a running average is used.

CONCLUSIONS

The model results indicate that it is possible to make a resonable representation of a precipitator without complex assumptions. The approach does gloss over much of the complexity of ESPs, particularly the electrical conditions, and does not address the causes of rapping losses. Still, it does a creditable job of estimating the effects of rapping losses.

The model emphasizes the overall importance of rapping emissions as a limit on the performance of ESPs. This emphasis is even stronger than in the static model previously developed. But the model also indicates a method for optimizing the rapper periods, which does correspond to actual practice in many cases (8).

There are many improvements possible in a simple model such as this one, and it is to be expected that the accuracy of the model would improve with them. But such improvements would not be expected to change the general nature of the results very much. Continued efforts along these lines should be very fruitful.

REFERENCES

1. White, H. J. Industrial Electrostatic Precipitation. Addison-Wesley, Reading, Massachusetts, 1963.

2. Oglesby, S., Jr., and Nichols, G. B. Electrostatic Precipitation. Marcel Dekker, New York, New York, 1978.

3. Lawless, P. A., Dunn, J. W., and Sparks, L. E. A computer model for ESP performance. EPA-600/9-82-005b. NTIS PB83-149-591. U. S. Environmental Protection Agency, Research Triangle Park, North Carolina, 1982.

4. Lawless, P. A. and Sparks, L. E. The importance of loss mechanisms in modeling electrostatic precipitator performance. EPA-600/7-79-044a. NTIS PB295-226. Paper presented at the EPA Fourth Symposium on the Transfer and Utilization of Particulate Control Technology, Houston, Texas. October 11-14, 1982.

5. Lawless, P. A. Electrostatic precipator research at Research Triangle Institute. Paper presented at the EPRI Conference on Electrostatic Precipitator Technology for Coal-Fired Power Plants, Nashville, Tennessee. June 1982.

6. Gooch, J. P. and Marchant, G. H. Electrostatic precipitator rapping reentrainment and computer model studies. Report No. EPRI FP-792. Electric Power Research Institute, Palo Alto, California, 1978.

7. Ensor, D. S., Lawless, P. A., Damle, A. S., and Sparks, L. E. Evaluation of the United McGill Electrostatic precipitator. EPA-600/9-82-005a. NTIS PB83-149-583. U. S. Environmental Protection Agency, Research Triangle Park, North Carolina, 1982.

8. Sproull, W. T. Minimizing rapping loss in precipitators at a 2000-megawatt coal-fired power station. JAPCA. 22:181, 1972.